ECTRACE - THREE DIMENSIONAL ACOUSTIC RAY TRACE PROGRAM

Robert Neal Christianson
Luis Ricardo Erazo

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ECTRACE - THREE DIMENSIONAL ACOUSTIC
RAY TRACE PROGRAM

by

Robert Neal Christianson
Luis Ricardo Erazo

March 1980

Thesis Advisor: A. B. Coppens

Approved for public release; distribution unlimited

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)

ECTRACE - THREE DIMENSIONAL ACOUSTIC RAY TRACE PROGRAM

5. TYPE OF REPORT & PERIOD COVERED

Master's Thesis;
March 1980

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)

Robert Neal Christianson

Luis Ricardo Erazo

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS

Naval Postgraduate School
Monterey, California 93940

12. REPORT DATE

March 1980

13. NUMBER OF PAGES

209

14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)

Naval Postgraduate School
Monterey, California 93940

15. SECURITY CLASS. (of this report)

Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

ASW officer student experience tour funding was provided by NRL, Washington, D.C. The initial background information and assistance was provided by Arctic Section, NRL (See Acknowledgement).

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acoustic Computer Model
Three-Dimensional Ray Trace
Acoustic Ray Tracing
Bathymetric Interference
Synthetic Ocean Bottom Model
Ocean Bottom Modeling
Bathymetric Curvature Effect

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Acoustic ray paths in the ocean are known to exhibit significant horizontal deflections after repeated reflections from the bottom. The effect may be quantitatively and qualitatively observed through a ray trace model which permits a change in direction of the vertical plane of propagation as a function of bottom slope and grazing angle. ECTRACE is a family of computer programs which traces a bundle of rays in three dimensions and utilizes bottom depth values as a portion of its input. Included are

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued)

related programs which develop sea bed models from digitized bathymetry data or synthetic bathymetry functions.

Approved for public release; distribution unlimited

ECTRACE - Three Dimensional Acoustic Ray Trace Program

by

Robert Neal Christianson
Lieutenant Commander, United States Navy
B.S., United States Naval Academy, 1970

Luis Ricardo Erazo
Lieutenant, United States Navy
B.S., Purdue University, 1972

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS TECHNOLOGY

from the

NAVAL POSTGRADUATE SCHOOL
March 1980

ABSTRACT

Acoustic ray paths in the ocean are known to exhibit significant horizontal deflections after repeated reflections from the bottom. The effect may be quantitatively and qualitatively observed through a ray trace model which permits a change in direction of the vertical plane of propagation as a function of bottom slope and grazing angle. ECTRACE is a family of computer programs which traces a bundle of rays in three dimensions and utilizes bottom depth values as a portion of its input. Included are related programs which develop sea bed models from digitized bathymetry data or synthetic bathymetry functions.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

This research was inspired by Dr. George Gielis and Mr. Charles Votaw of the Arctic Section, Acoustic Division of the Naval Research Laboratory, Washington, D.C. We thank these gentlemen and their assistants for devoting considerable time and effort to our endeavors.

A special thank you for NRL in supporting the ASW student officer experience tour (six weeks) which allowed us to gain the expertise needed to attempt this thesis.

I. INTRODUCTION

ECTRACE is a FORTRAN computer program designed for the investigation of bathymetric effects on horizontal acoustic ray tracing. Its supporting computer programs include routines for constructing a grid model of the sea bed from contour data, ray tracing routines, and associated printed and plotted output. The programs were written for IBM 360/370 series systems with a Versatec plotter and related software. The significant feature of the tracing algorithm is its ability to shift the direction of the vertical plane of propagation after a bottom reflection. A sea bed modeled by triangular facets fitted through three points of a grid cell makes this feature possible with a minimum effect on computation time.

Several models for tracing acoustic rays in the ocean are used in the Navy today. These models are based on a simplified environment because of practical limitations on data and computer run time, and they adequately trace a bundle of rays in a single vertical plane. Some enable intensity calculations at a given distance from the source. TRIMAIN [1], for example, is a model which features range as well as depth dependence of sound speed and accounts for some interaction with irregular bathymetry. These models have a large number of applications in deep ocean cases involving ducts, channels, and convergence zones, and bottom reflections where the plane of the ray path does not exhibit horizontal deflection.

Actual cases of long range propagation over irregular bottoms show any single ray path rarely remains within a single vertical plane. Any application of ray theory to propagation involving bottom interaction must account for a horizontal deflection caused by ray heading changes after repeated reflection from a tilted bottom plane. This effect is most pronounced for long range paths over smoothly sloping surfaces, but any valid ray approximation can be seen to undergo significant heading changes after only a few reflections from an irregular or undulating bottom.

The behavior of sound propagation over troughs, ridges, wedges, and seamounts has been studied analytically by Harrison [2] using normal mode theory and ray invariants. Since actual ocean bottom topography over large area defies practical analytical description, except in the stochastic sense, DeWitt [3] developed numerical techniques for three-dimensional ray tracing in which the bottom consists of triangular facets generated numerically from actual or interpolated bathymetry data. This approximation scheme greatly simplified the problem of calculating the three-dimensional ray path parameters after a bottom reflection and provides a means of constructing reasonably accurate ray traces (for a verification, see Appendix F).

The ECTRACE modeling program is an adaptation of the Dewitt technique for use on an IBM 360/370 series computer. The initial programs may be used to generate synthetic bottom topographical grids or create approximations to real ocean bottom grids from a bathymetric data source file. The

topographical features of the generated grid can be depicted by a contour plot and a perspective surface plot for comparison with the original contours. The primary program is the ray tracing program which accepts the generated bottom grid and performs stepwise ray position calculations from user-selected initial heading and elevation angles. The output is the printed history of each ray, and a two-dimensional plot of the ray paths projected on a vertical plane intersecting the bottom profile along the axis of the ray fan. Additionally, a horizontal plane projection is plotted which may be overlaid on the appropriate contour plot to correlate the horizontal curvature effect with grid bathymetry. Punched card output from several ray tracing runs may be combined in one independent routine for a comprehensive visual inspection of acoustic shadow zones in the horizontal plane.

It is very important to note that ECTRACE permits refraction caused by sound velocity gradients in the vertical plane only and that the sound velocity in the horizontal plane is assumed constant within a well defined water mass. All horizontal bending effects revealed by ECTRACE are the result of bottom reflection only.

With the ability to convert actual bathymetry data to a matrix of depth values, discrete faceted approximations of a selected oceanic region can be used for generating ray traces of operational significance. In addition, qualitative studies of horizontal ray deflections over idealized bottom configurations are enhanced by the computer-generated plots of ray paths over easily modifiable synthetic bathymetry. ECTRACE and the

augmenting programs are envisioned to serve as a basis for future development of three-dimensional ray modeling and investigation efforts.

II. DESCRIPTION OF ECTRACE AND PERIPHERAL PROGRAMS

This chapter describes each of the programs, their primary subroutines, and their interrelationship in producing a ray trace which may be used to investigate the bathymetric effects on horizontal ray curvature. They are presented in their intended order of use. Flow Diagram (Fig. 1) and Table I are provided to assist in understanding program relationships.

A. GENBOT

This program converts contour data into a matrix of depth values for storage on a permanent device. It also makes a two dimensional contour plot of the input data and superimposes reference latitude and longitude lines (Fig. 2). The contour plot can be placed underneath the horizontal ray trace plot of ECTRACE or ECCOM on a light table to aid in the visual investigation of the ray curvature (see Figs. 3 and 4, and Figs. 5, 6, and 7).

The bathymetry contour data used in this research exists as a sequence of data sets representing ten by ten degree regions of the North Atlantic Ocean and were obtained exclusively from a tape compiled in a joint project of the Naval Research Laboratory and the Mobil Oil Company. In their original form the data were sufficient only for producing a computer plot of region contours, thereby reproducing portions of a chart titled "Bathymetry of the Norwegian-Greenland and Western Barents Sea" [4] (see Fig. 8). Their use

in GENBOT required us to insert the depth values of the contour lines and transfer the results to a separate tape (described in Appendix A).

The following subroutines are called by GENBOT:

1) CORNER calculates boundary latitudes and longitudes for screening only those data needed for interpolation.
2) GEOPLT superimposes reference latitude and longitude lines on the subregion contours.
3) GEODST calculates the forward or inverse solutions of the geodetic triangle to accomplish the above.
4) RAIN 1, RAIN 2, and RAIN 3 [5] were obtained from the SSP3 program library at NPS and are used to perform interpolations on data points for construction of the depth matrix.

B. SYNGEN

This program produces a synthetic bathymetry grid for storage on a permanent device. It may be used to generate the following types of ideal sea bed configurations: wedge, trough, ridge, conical seamount (Fig. 9), sinusoidal undulation, or parabolic basin (Fig. 10). These grids can be used in the ECTRACE program as a depth matrix in the same manner as the grids of actual bathymetry produced by GENBOT. SYNGEN is discussed in detail in Appendix B.

C. G3DP

Program G3PD produces a perspective of any portion of the generated depth matrix through use of the NPS system subroutine CONTUR [4]. Figures 9, 10, and 11 are examples. G3DP uses

job control commands to link with the GRDSCT subroutine of the ECTRACE load module to extract the desired area (hereafter called the working matrix) to be plotted. Appendix C describes the application of G3DP in detail.

D. GRDCHK

This program is used to check the validity of the depth matrix generated by the programs GENBOT and SYNGEN. It produces a contour plot and a vertical plot of user-selected rows of the generated matrix. By comparing consecutive rows the existence and location of anomalous data points may be determined (Figs. 4 and 12). It contains the previously mentioned subroutine GEODST, CORNER, and a version of GEOPLT (called GEOPLR) to draw reference geographic lines for comparison of the two contour plots (GENBOT vs GRDCHK). Refer to Appendix D for additional discussion of GRDCHK.

E. ECTRACE

The ECTRACE program is a package of subroutines which exist in load module form on a permanent storage device in the NPS computer system user library. Its use requires a user supplied calling program which calls subroutine TRACER and contains the necessary job control language (JCL) to link with the module. These procedures are explained in more detail in Chapter IV of this report.

ECTRACE traces rays in three dimensions. The sound speed field is assumed to vary piecewise-linearly with depth only, yet provisions are made to permit simulation of up the three distinct water masses separated discontinuously by vertical

fronts. The bathymetry structure, generated by one of the two grid-making programs, is a three dimensional surface modeled by discrete triangular facets fitted through cells of adjacent depth values. Rays are traced as piecewise arc segments each with a radius of curvature dependent on the vertically-structured sound speed gradient.

When the rays undergo a bottom bounce, specular reflection is assumed. Bottom loss values can be calculated from the subprogram FBLOSS (described later), a standard Navy model such as the FACT bottom-loss routine, or a user supplied routine. The trace history supplied by ECTRACE includes the location and characteristics of each ray reversal (surface reflections, bottom bounces, and refractive turning points). The printed data lists the new ray parameters determined after the reversal had occurred. Chapter V explains the printed output in more detail.

The ECTRACE plot product includes a plane view of the ray paths projected on a vertical plane, a plot of sound speed profiles, and a horizontal (plane) view of the ray paths (Figs. 5 and 13). The horizontal view can be overlaid on the appropriate contour plot generated by GENBOT or GRDCHK, as in Figs. 12 and 14.

The following subroutines are used by ECTRACE:

1) Tracer is the primary tracing subroutine which calls all other subroutines of the ECTRACE program and is capable of processing a bundle of rays separated into ray fans. A ray fan is defined here as a number of

rays (user determined) each with a different initial heading measured clockwise from grid north, but all with the same initial elevation angle. A positive elevation angle describes a ray which is pointed toward increasing depth.

2) GRDSCT reads into core a fixed-dimension section of a depth matrix file of arbitrary size. The ray trace is restricted to the boundaries of the core-loaded working matrix. A ray which departs the working matrix boundaries is terminated and program control is passed to the next ray's trace.

3) IDTSUB identifies the three matrix subscript pairs whose depth values define a plane triangular bottom facet underneath the ray segment head.

4) DEEP calculates the depth at any horizontal position by solving for the z-coordinate on a plane fitted through the three depth matrix values.

5) CONTAC calculates the three-dimensional coordinates of the intersection of the ray path and triangular bottom facet.

6) BOANG calculates the grazing angle, elevation angle, and new heading of a bottom reflected ray.

7) FBLOSS returns bottom loss values in dB as a function of grazing angle for a bottom reflected ray. These values are based on NRL geophysical survey data of the Greenland-Norwegian Sea and Iceland environment collected in the mid-1970's and have been widely averaged for

convenience. The user has the option of substituting a more specialized function if bottom loss values are critical.

8) IDPROF determines the water mass, and hence the sound speed profile, affecting the ray with each segment iteration.

9) NUPROF initializes calculations for new water mass parameters once the ray has passed the boundary.

10) CHNLIM determines in advance the refractive sound channel limits for each ray from its invariant. This is used to test a ray at its turn-around point and to reveal its type (surface reflected, refracted-surface reflected, bottom reflected-surface reflected) in the printed trace history.

11) ANGPRT is used to print a summary of selected bottom contact in parameters for each ray fan.

12) The following subroutines each have only one calling statement in TRACER. The subroutines they in turn invoke are dependent upon the computer graphic system installed and would require internal modification if exported to a system without Versatec software. For additional information NPS users should consult Ref. 7.

    a) BGNPLT initializes the plotter and draws all borders, titles, and axes.

    b) BDTPLT draws a vertical profile of the sea bed centered along the mean heading of the ray bundle, as shown in the upper plot of Figure 13.

c) RNGPLT traces all ray segment projections on the vertical plot.

d) T2DPLT plots the horizontal track of each ray. Each ray fan uses one of thirteen symbols to indicate the bottom bounce positions of each ray. Figure 5 shows some of the available symbols.

e) SSPPLT draws the sound speed profiles.

f) ENDPLT draws the horizontal plot legend and terminates all plotting.

F. ECCOM

Program ECCOM uses the optionally produced punched card output of one or more ECTRACE jobs to draw a composite horizontal plot of several ray bundles, enabling a comprehensive pictorial study of shadow zones and acoustic convergence in the horizontal plane. The composite horizontal plot is designed to portray the rays which emanate from the same source or from multiple sources along a desired track. The ray numbers assigned and punched for each ray on each ECTRACE run assist the user in singling out rays of interest (or non-interest) before combination. Figures 7 and 18 are examples of this combining technique.

Amplifying remarks on these programs and subroutines can be found in the comments in the computer FORTRAN listings. For additional description of plots and computer printed outputs, refer to the appropriate appendices.

III. PROGRAM LOGIC AND THEORY

ECTRACE traces the three-dimensional paths of a bundle of rays, one ray at a time. The bundle is divided into fans of rays of the same initial elevation angle. A separate ray fan is traced for each fixed increment of initial elevation angle between the limits ELST and ELEND (input constants). A positive elevation angle describes a ray pointed downward. The initial ray headings in each ran fan extend from HDST to HDEND (input constants), measured in degrees clockwise from grid north.

A. BOTTOM AND SOUND FIELD MODELING

Figure III.1 shows the projection of the bottom surface onto the horizontal x-y plane.

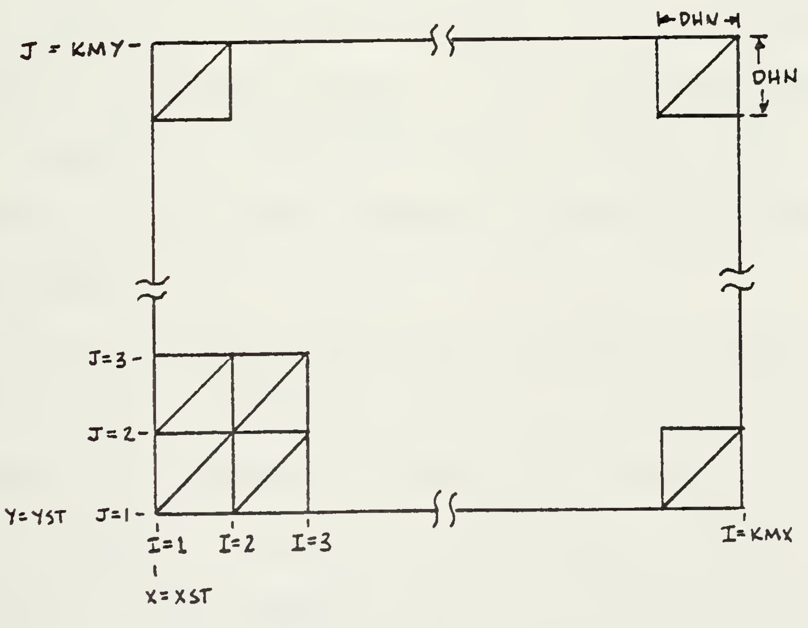


FIGURE III.1 Bottom Surface Projection

The projection of each triangular facet is a right isosceles triangle. The entire surface can be defined by specifying the depth values at the vertices of the projection triangles. ECTRACE stores these depth values (km) in the two-dimensional working array ZB. The parameter DHN (km) represents the spacing between the vertices in both the X (East) and Y (North) directions. Thus, the value of the matrix point ZB(I,J) is the depth at the vertex point with X coordinate (I - 1) * DHN and Y coordinate (J - 1) * DHN, relative to the ZB origin.

The matrix values for ZB are read from a source file stored on a permanent storage device such as a disk pack. A matrix of dimensions 151 by 151 with spacing DHN = km has been used for trial runs.

1. Bottom Contact Point

The subroutine CONTAC locates the coordinates and calculates the depth of the bottom bounce point. The parameters describing locations of the ray segment head and tail are among those passed to the main program. In this subroutine, the ray segment is treated as unrefracted (a straight line), a reasonable approximation for the small distances involved in determining the contact point.

First the horizontal positions of the ray segment head (x,y) called CEE and CNN in CONTAC, and (x_1, y_2) and called CRE and CRN in CONTAC, are checked to see if they both lie in the same projection triangle. If not, an interative routine searches for two points on the segment such that the horizontal coordinates of both points lie inside the same triangular cell and

define the portion of the segment that penetrates the bottom facet. Then the depth of the ray head z, and the depth of the tail z_1 are computed.

Figure III.2. illustrates a profile of a cell bottom with a ray intersecting it in the propagation plane. The solution of the triad (x_c, y_c, z_c) identifying the point of bottom contact proceeds as follows:

Using subroutine DEEP (described in Section 3) compute H_1 and H_2, the bottom depth at (x_1, y_1) and (x_2, y_2), respectively. Then compute:

$$\Delta z = z_1 - z_2$$

where (x_c, y_c, z_c) are contact point parameters.

$$z_c = \frac{H_1 z_2 - H_2 z_1}{\Delta z + H_1 - H_2}$$

$$\text{If} \quad \Delta z \neq 0, \; M = \frac{z_c - z_1}{\Delta z}$$

$$\text{If} \quad \Delta z = 0, \; M = \frac{z_1 - H_1}{H_1 - H_2}$$

$$\text{Then} \quad x_c = x + M\, r \sin \phi \quad \text{and}$$

$$y_c = y + M\, r \cos \phi$$

where ϕ is the ray heading (angle between the y-z plane and the propagation plane).

2. Projection Triangle Vertex Subscripts

The subroutine IDTSUB calculates the I subscripts, IRT, ILT, IVR and the J subscripts JBT, JTP, JVR for the vertices of

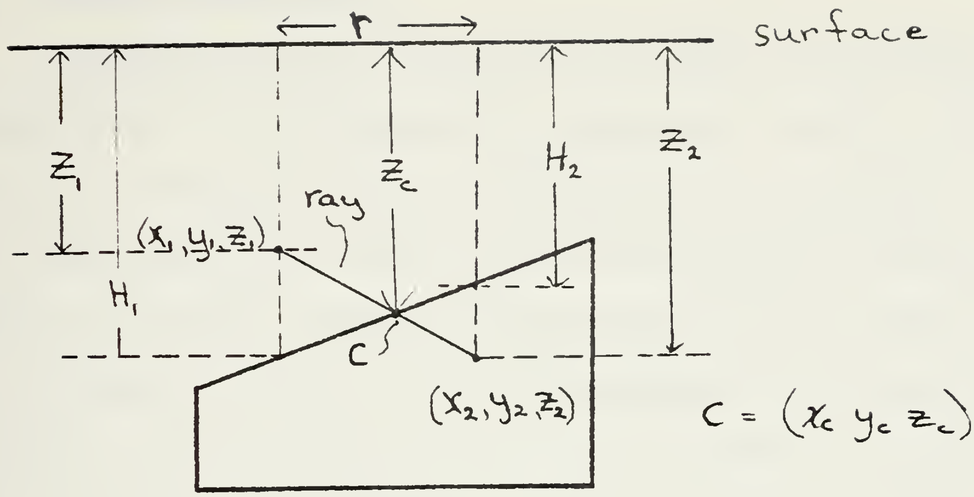


FIGURE III.2. Profile of the bottom cell in the propagation plane, with a ray contact.

of the projection triangle which contains the ray's horizontal coordinates x and y. The subscript parameters are indicated in Figure III.3. The following suffixes identify the relative locations of the vertices of the projection triangle:

RT - right

LT - left

TP - top

BT - bottom

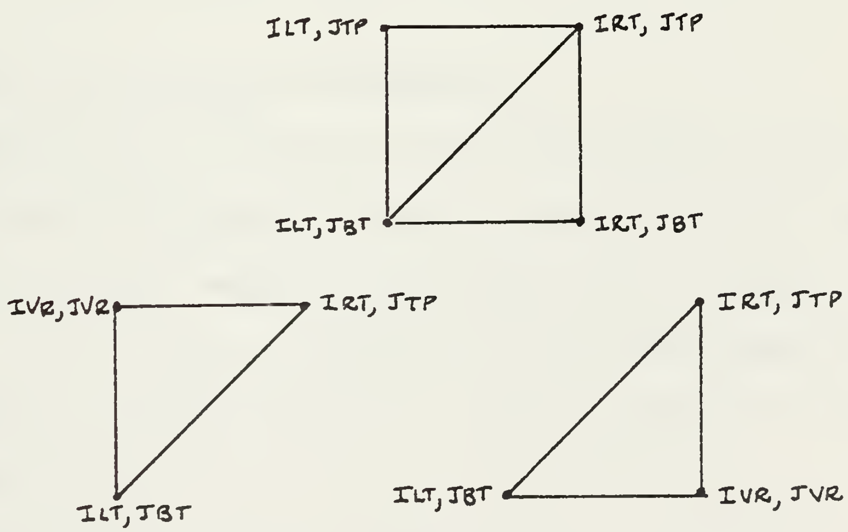


FIGURE III.3. Projection Triangle Vertices

3. Depth Calculation

Subroutine DEEP calculates the depth in kilometers corresponding to the horizontal coordinates x and y. Automatically, IDTSUB is called to identify the projection triangle subscripts. The depth z_c is calculated by solving the equation of a plane defined by the depths at the vertex points (D_1, D_2, and D_v). Figure III.4 illustrates the calculations made based on the orientation of the projection triangle outlining the bottom facet plane. The quantities x_c and y_c represent the horizontal position (in units of DHN) of the depth calculation point relative to the horizontal position of the depth calculation point relative to the horizontal position of the depth value D_1 (the lower left vertex of the projection triangle) as calculated in Secion 1. The depth values D_x and D_y are normalized by the length of the equilateral side l, in units of DHN. Finally, the depth is solved by

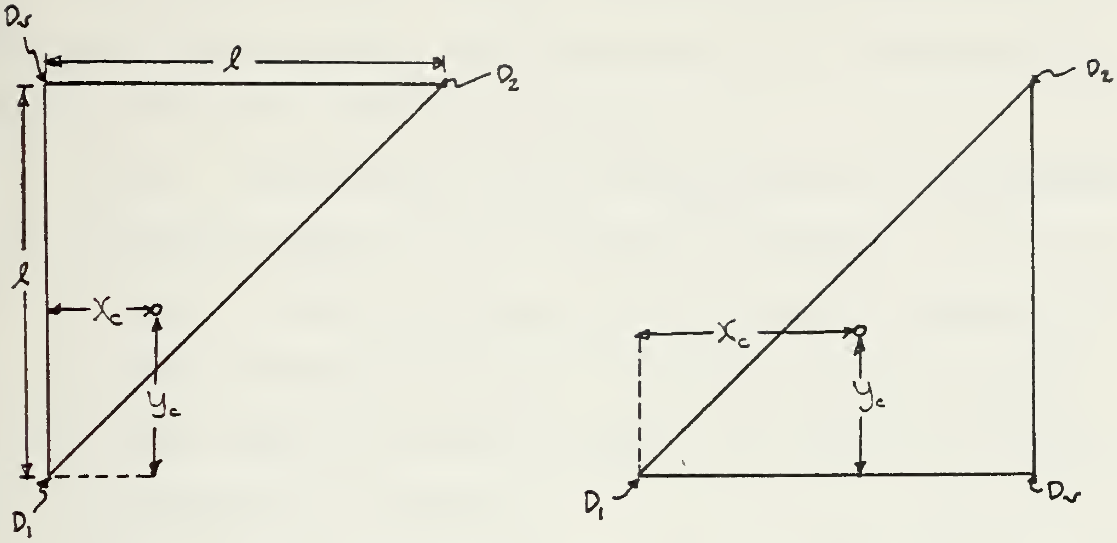
$$z_c = x_c D_x + y_c D_y + D_1$$

B. RAY LOCATION AND DIRECTION PARAMETERS

A three dimensional left hand coordinate system is used (Fig. III.5.), with the positive z axis pointing in the direction of increasing depth and the sea surface level lying in the x-y plane. As the ray is traced through three dimensional space, the program variables which are used to locate the tail of the ray vector are:

DEP1 - depth coordinate

CRE - x coordinate in km



$$D_x = \frac{D_2 - D_v}{\ell} \qquad D_x = \frac{D_v - D_1}{\ell}$$

$$D_y = \frac{D_v - D_1}{\ell} \qquad D_y = \frac{D_2 - D_v}{\ell}$$

FIGURE III.4. Depth calculation on bottom facet

CRN - y coordinate in km

CRE and CRN are components of the horizontal range increment DR which forms the propagation plane heading angle PHI with the y-axis.

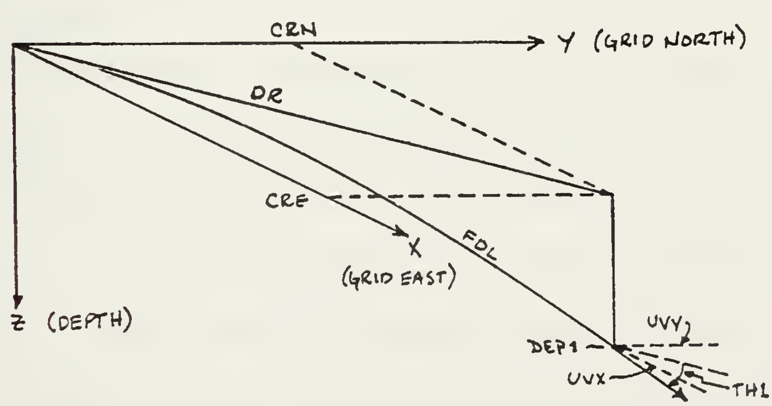


FIGURE III.5. Ray Location Parameters

Given that the ray vector originates at the point with the above coordinates, the direction of the ray vector is specified by the following angles:

TH1 - ray elevation angle in the plane of propagation measured from the ray vector's projection on the xy plane to the ray vector. A ray pointing toward the ocean bottom will have a positive elevation angle. Theta takes on values between $-\pi/2$ and $+\pi/2$ (output values are in degrees).

PHI - ray (propagation plane) heading angle between the positive y-axis and the projection of the ray vector on the xy plane, measured clockwise from the positive y-axis (grid North). PHI takes on values between $-\pi$ and π.

Initial values for TH1 and PHI are calculated from user input values (in degrees), which are converted to radians for computational purposes. The unit vectors in the horizontal plane are given as UVX = sin (PHI) and UVY = cos (PHI).

C. RAY PATH CALCULATIONS

1. Path Length

ECTRACE confines refraction to the vertical plane alone. It is recognized that while horizontal sound speed gradients exist, their effects are sufficiently small within a well defined water mass to be neglected compared to the bathymetric effect.

However, while the horizontal gradients are likely to be slight, important discontinuities in water characteristics may occur horizontally across oceanic fronts. For this reason,

ECTRACE has been designed to simulate up to three distinct water masses and two associated frontal boundaries.

Within a water mass having uniform characteristics along fixed depths a ray path conveniently defines a vertical plane within which Snell's Law can be written as

$$V = c(z)/\cos\theta(z) \qquad (1)$$

where V is treated as invariant within the water mass (termed the vertex velocity in some texts since it is the speed at the depth at which the ray vector becomes horizontal). The term $c(z)$ is the sound speed at depth z and $\theta(z)$ is the elevation angle at depth z.

Equation (1) uses the simplification that the speed of propagation and the elevation angle are functions of depth only. It is generally possible to define a stratified medium in which each layer's gradient is constant. Although such a model requires a large number of data points to adequately describe a water column of complicated sound speed structure, the piecewise linear gradient approximation greatly simplifies the ray path descriptions.

Taking advantage of the approximations, we use a simple equation for the sound speed gradient, within a layer,

$$g = dc/dz \qquad (2)$$

Integrating yields

$$z = c/g \qquad (3)$$

where the constant of integration is avoided by arbitrarily specifying the origin of the coordinate axes at the depth where $c = 0$. By applying Snell's Law we obtain

$$z = \frac{V}{g} \cos\theta \tag{4}$$

and

$$dz = -\frac{V}{g} \sin\theta d\theta \quad . \tag{5}$$

Specifying r as the horizontal distance axis of the vertical plane and applying analytical geometry there results

$$dz = \tan\theta dr \tag{6}$$

$$dr = -\frac{V}{g} \cos\theta d\theta \tag{7}$$

and

$$r = -\frac{V}{g} \sin\theta \tag{8}$$

when measuring r from the vertical (z-axis) to the point at which $\theta = 0$ (horizontal projection).

Squaring (4) and (8) and summing yields

$$r^2 - z^2 = (\frac{V}{g})^2 \tag{9}$$

This is the equation of a circle of radius (V/g) and whose center is at the origin of the coordinate system just described. The equation

$$R = -\frac{V}{g} \tag{10}$$

describes the radius of curvature of a ray path in a constant

sound speed gradient. The minus sign has been chosen to allow
to be positive when the refracted ray is increasing its elevation angle.

The coordinate system and computer-approximated ray path segments are shown in Fig. III.6.

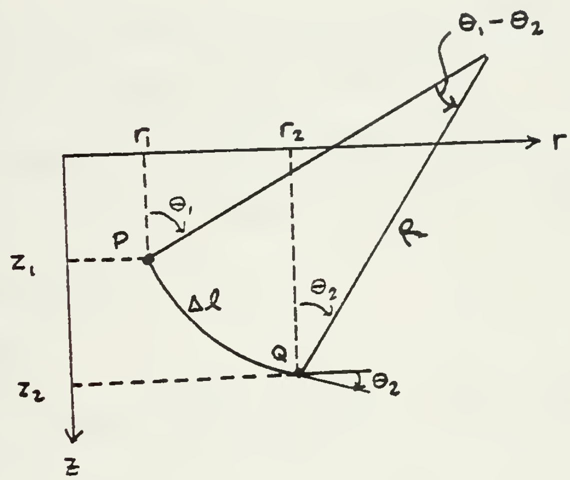


FIGURE III.6. Ray Path Geometry

It can be shown from Fig. III.6. that the ray segment path length between P and Q is

$$\Delta \ell = \rho \Delta \theta \tag{11}$$

where

$$\Delta \theta = \theta_2 - \theta_1 \tag{12}$$

Since the initial elevation angle of the ray is presumed known, successive values of θ are found iteratively:

$$\theta_2 = \theta_1 + \frac{\Delta \ell}{\rho} \tag{13}$$

mple geometry also allows for solution of the depth change,

$$\Delta z = z_2 - z_1 = \rho(\cos\theta_1 - \cos\theta_2) \quad , \tag{14}$$

d likewise the increase in horizontal range,

$$\Delta r = r_2 - r_1 = \rho(\sin\theta_2 - \sin\theta_1) \quad . \tag{15}$$

ote that in the Fig. III.6., $\Delta\ell$, Δz, and Δr must all be positive, o the sign of ρ must agree with that of $\Delta\theta$. Equation (10), hich gives ρ the opposite sign of the gradient, satisfied this equirement. Only Δz is allowed to become negative, as when a ay decreases its depth.

2. Travel Time

From the diagram (Fig. III.7.) the sound speed relationship becomes

$$c = \frac{d\ell}{dt} \tag{16}$$

$$dt = \frac{dz}{c\sin\theta} \tag{17}$$

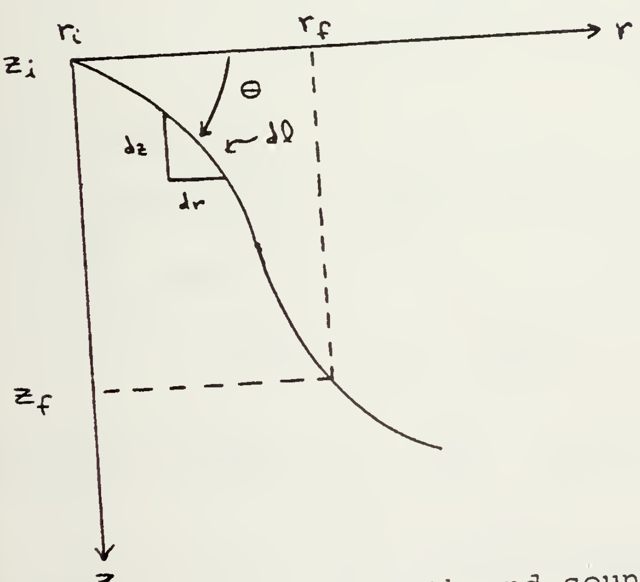


FIGURE III.7. Ray path and sound speed relationship.

The ray invariant

$$a = \frac{1}{V} = \frac{\cos\theta}{c} \tag{18}$$

and the trigonometric identify $\sin\theta = \sqrt{1 - \cos^2\theta}$ combine and give dt as a function of depth

$$dt = \frac{dz}{c(t)\sqrt{1 - [ac(z)]^2}} \tag{19}$$

The time-of-flight of the acoustic ray along the path is found by integrating from the initial point to the final point:

$$t_f - t_i = \int_{z_i}^{z_f} \frac{dz}{c(z)\sqrt{1 - [ac(z)]^2}} \tag{20}$$

Before integrating it is assumed that the gradient is constant along the segment. More generally, ECTRACE assumes a constant gradient in the layer bounded by z_1 and z_2.

Then

$$c(z) = c(z_1) + g(z_2 - z_1) \tag{21}$$

where $z_1 \leq z \leq z_2$.

The integral may now be evaluated by introducing a new variable $W = \frac{c(z)}{g}$ such that

$$W = z - z_1 + \frac{c(z)}{g} \tag{22}$$

Then $dW = dz$ and $C(W) = gW$. Therefore,

$$t_f - t_i = \int_{W_i}^{W_f} \frac{dw}{gW\sqrt{1 - a^2g^2w^2}}$$

$$= \frac{1}{g} \ln \left\{ \frac{W_f [1 + \sqrt{1 - a^2 g^2 Wc^2}]}{W_i [1 + \sqrt{1 - a^2 g^2 Wf^2}]} \right\} \quad (23)$$

$$= \frac{1}{g} \ln \left[\frac{W_f \ (1 + \sin\theta_i)}{W_i \ (1 + \sin\theta_f)} \right] . \quad (24)$$

Values for W are restricted to being in the same layer and on the same side of a turning point in the ray path.

D. RAY TRACING PROCEDURE

ECTRACE begins tracing a ray from its given initial coordinates CRE, CRN, and DEP1. First the water mass is identified to determine the local gradient and therefore the ray invariant and radius of curvature. The ray path is constructed in the form of arc segments using an increment DTH calculated from a chosen arc length DL. Then the new θ value is calculated and Equations (14) and (15) are used to determine the ray segment's depth increment and horizontal range increment. The horizontal position is updated using the horizontal unit direction vectors (UVX and UVY) calculated from PHI, the initial heading of the ray's plane of propagation.

The interim update 3-D coordinates of the ray head may now be stated in FORTRAN as

CEE = CRE + DR * UVX (25)

CNN = CRN + DR * UVY (26)

DEP2 = DEP1 and DZ (27)

where DR represents the horizontal range increment Δr and DZ is the depth increment Δz.

E. RAY CONSTRAINT CONDITIONS

Each time the ray path is incremented, certain constraint conditions must be checked. These constraints are bottom reflections and the sound channel limits.

1. Bottom Reflections

After the interim horizontal coordinates CEE and CNN are calculated, they are passed to DEEP to find the bottom depth WH2 at the ray head. If DEP2 is greater than or equal to WH2, a bottom bounce has occurred. In this case the values of DEP2, CEE, and CNN are adjusted by linear interpolation and a point of bottom contact is established. Subroutine BOANG is then called to calculate new values of TH1, PHI, UVY, and the ray grazing angle GRAZ.

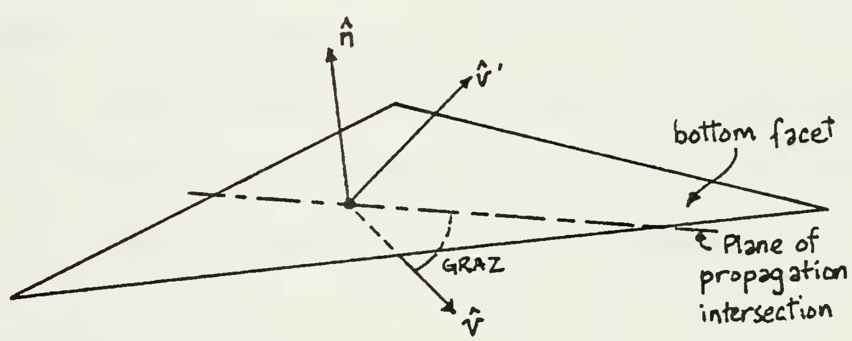


FIGURE III.8. Bottom Bounce Reflection Angles.

The basic calculations of BOANG involve the vectors:

$\hat{v}$ - unit ray vector before bounce

$\hat{n}$ - unit vector normal to triangle facet at the bounce point

$\hat{v}'$ - unit ray vector after bounce

The components of $\hat{v}$ are calculated from TH1 and PHI. The components of $\hat{n}$ are calculated from the equation of the plane of the bottom facet. The new vector $\hat{v}$ is then calculated from

the vector equation,

$$\hat{v}' = \hat{v} - 2\ (\hat{v} \cdot \hat{n})\ \hat{n} \quad . \tag{28}$$

New values of TH1, PHI, UVX and UVY are then calculated from the compnents of $\hat{v}'$. Finally, the grazing angle GRAZ comes from

$$\sin(\mathrm{GRAZ}) = -(\hat{v} \cdot \hat{n}) \quad . \tag{29}$$

Reflection from a sloping bottom plane facet, as opposed to a surface reflection, forces a reevaluation of the Snell's Law constant V, and therefore the radius of curvature R. The invariant V remains in effect until the next bottom bounce occurs or a new water mass is entered. Thus, after summoning BOANG, the ray tracing continues with the new ray parameters. The sound channel limits CHS and CHD described below are also recalculated at this point since they also depend on V, and GRAZ is passed to function FBLOSS to calculate a bottom loss value.

2. Sound Channel Limits

The Snell's Law equation (1) says that the ray trace is constrained to depths where the speed of sound does not exceed V. A given ray path may be constrained in the sense that it is purely refracted (R), refracted or reflected (RSR or RBR) or purely reflected (BRSR). The determining factors are V and the local sound speed by profile. V is calculated from the sound speed at the initial depth and the initial elevation angle, and is recalculated after a bottom bounce or upon entering a new water mass.

CHS and CHD are the refractive sound channel limits at the shallow and deep extents respectively. Since the sound speed profile is approximated by piecewise continuous linear gradients, the water column is treated as a column of indexed layers, each characterized by upper and lower (shallow and deep) depth boundaries and a constant sound speed gradient. The limiting depths CHS and CHD are found by scanning the water mass layers both above and below the current ray location until a boundary sound speed value is found which exceeds V. The limiting depth is then calculated from

$$z_{lim} = z + (V - c)/g \qquad (30)$$

where z and c are evaluated at the top of the layer and g is the gradient of the limiting layer.

For cases where g = 0 or where V is not exceeded by any sound speed in the local profile, CHS and CHD are set to the physical limits of the water column.

F. RAY PROPAGATION TERMINATION CONDITIONS

Tests are made during each ray segment iteration for events which terminate the ray trace. These events are:

- Escape from grid - when the ray departs the finite x and y constraints of the working depth array (ZB).
- Depth boundaries - when the depth detected at the ray is less than ten wavelengths or greater than 10 km.
- Bottom loss - when the attenuation due to bottom reflection exceeds a specified input amount in dB.

- Error at bottom contact – when defective program code or anomalous depth data prevent the interpolative routine CONTAC from recovering the bottom depth within a reasonable number (50) of iterative attempts.
- Trapping - when it becomes apparent that the ray will not contact the bottom, rendering further three-dimensional tracing unnecessary.
- Bottom bounce gate - when the total number of bottom reflections exceeds a user specified amount.

The cause of ray termination is specified in the printed ray history, giving the ray parameters at the time of termination.

G. IMPROVEMENTS MADE OVER THE ORIGINAL NRL VERSION

The ray tracing algorithm originally existed as a program called ABOUNC at the Naval Research Laboratory. ECTRACE differs significantly from ABOUNC in the following areas:

1. Ray Path Development

ABOUNC calculates ray parameters through right triangle solutions and Snell's Law, but develops the ray path along fixed depth. Since this prevents rays from becoming horizontal, a reversal is forced when the magnitude of the elevation angle goes below a gate value. In contrast, ECTRACE develops the ray path along arc segments and radii of curvature, fixing only the maximum length of the segments while interrupting the increment as necessary when a segment reaches a reversal point or layer boundary, or to keep the increment within a maximum amount.

2. Computer Word Storage

ABOUNC floating point variables are nearly all FORTRAN double precision. The largest single burden on computer core is the depth matrix, read from a magnetic tape in card image format, each card record containing eight items (fields) ten bytes long. This format permitted a depth range of ±9999 km with resolution of ±1 cm. By reducing the depth range to ±999 km and resolution to ±10 cm, two unnecessary bytes are removed from each field, enabling a fit of ten items per record and a permanent storage reduction of twenty percent. In addition, ECTRACE reads the matrix into a single precision array to reduce the core requirements for the data by fifty percent.

3. Grid File Manipulation

ABOUNC reads the contents of the entire depth matrix data file into array whose dimensions must be exactly equal to those of the matrix. This requirement is not inconvenient on a large computer with FORTRAN dialect which permits objects time dimensioning. ECTRACE is written in FORTRAN G for an IBM Series/360 Model 67 machine whose limitations required some fundamental modifications of the program and its job control.

To maintain flexibility and reduce run time, ECTRACE was reduced to a package of subprograms which have been precompiled and stored on a permanent device in load module form. The user must supply a calling program which serves primarily to establish the dimensions of the depth array. The user's selection of array dimensions are made on the basis of core economy and area of interest rather than the dimensions of the

input matrix. ECTRACE, through its internal subroutine GRDSCT, extracts a rectangular working subregion from any part of the input matrix. Trial runs of ECTRACE used a working array of dimensions 151 by 151 representing a sector of 150 km on a side with a core requirement of less than 250K, while the input matrix was 369 by 443. Without GRDSCT, ECTRACE would have required an allocation of 800K bytes using the same input file.

4. Treatment of Small Elevation Angles

As stated previously, ABOUNC forced a reversal of the ray as the elevation angle of the leading segment approached a gate value close to zero. Since the ECTRACE tracing scheme permits a full range of elevation angles, early trial runs revealed an inherent error phenomenon as θ approached zero in a steep gradient. Specifically, a $\Delta\theta$ on the order of 3 degrees caused by a small radius of curvature or along arc segment traced a ray to its vertex depth in error as much as 50 meters from the Snell's Law prediction. Since the depth change was calculated from the difference in elevation angle cosines, the error was found to arise from the machine inability to retain difference precision for any pair of numbers close to integer values (round off error). Since the difference rather than the actual cosine values is needed for this calculation, the small angle series approximation cos $1 - \frac{\theta^2}{2} + \frac{\theta^4}{4!} - \ldots$ was used to formulate the equation

$$\cos\theta_1 - \cos\theta_2 \cong \tfrac{1}{2}(\theta_2^2 - \theta_1^2) - \tfrac{1}{24}(\theta_2^4 - \theta_1^4). \qquad (31)$$

When restricted to small angles, this approximation yields

greater accuracy than the straightforward machine calculation of the left hand side, since the precision retained by the registers is very high.

5. Travel Time

ABOUNC calculates the travel time of an acoustic signal along a ray segment by dividing the linearly-approximated path length by the mean sound speed. ECTRACE accounts for path curvature and depth dependence of the sound speed through integral evaluation, discussed in detail in section C.2.

6. Track Events

Layer transitions, reflections, and turnarounds are events which may require a change in ray parameters. ABOUNC tests for these events in algorithmic sequence, makes the necessary changes in response to the first event detected and continues the trace from the current position of the ray vector head. ECTRACE retraces a ray segment to the event depth and tests for all other events before beginning the next segment from the transition point. This logic eliminates the possibility of a long ray segment crossing more than one event depth with only one event detected.

IV. USER INSTRUCTIONS (ECTRACE)

A single step ECTRACE job traces a bundle of rays from a single source in the form of ray fans. Each ray fan is a bundle of rays of common initial elevation angle. All ray fans are bounded between to initial heading values measured from grid North (grid North equals true North at the grid center).

During the following discussion the reader may wish to refer to Fortran listing in Appendix M. The user instructions for the other programs (GENBOT, G3DP, SYNGEN, GRDCHK, and ECCOM) are explained in their individual appendices (User Instructions and Output Description).

A. JOB CONTROL CARDS (JCL)

The following JCL cards and parameters are required for an ECTRACE run:

- JOB card - The CPU time parameter should allow 60 seconds for every seven rays to be traced.
- EXEC card - This card must specify the FORTCLGW procedure (Appendix O). 250K bytes of core is required for the standard 22801 (151 by 151) element depth matrix portion plus 4K for every additional 1000 elements. The computer systems job class definitions should be considered before deciding upon the size of the working matrix and the number of rays. For example,

  EXEC FORTCLGW,REGION.GO=250K

  is a class C job at NPS when limited to five minutes of

time, which is adequate for tracing 40 rays.

- Calling program - Following the

  ```
  //FORT.SYSIN DD *
  ```

  card, the calling program must contain cards

  ```
  COMMON/DIMS/KMX,KMY and
  CALL TRACER (ZB) .
  ```

  ZB is a REAL * 4 array dimensioned exactly by KMX,KMY, and these values (KMX,KMY) are initialized to the values of the DIMENSION statement using assignment statements, as in the following example.

  ```
  DIMENSION ZB(151,151)
  COMMON /DIMS/ KMX,KMY
  :
  KMX=151
  KMY=151
  :
  CALL TRACER(ZB)
  :
  STOP
  END
  ```

  KMX and KMY are INTEGER * 4 and should be less than or equal to the dimensions of the data matrix from which the depth values will be taken.

- LINK cards - These cards point to the ECTRACE load module. They must include all members of the module for which the user does not supply a substitute. An example which includes all members is:

  ```
  //LINK.USDD UNIT=3330,VOL=SER=DISK01,DSN=1270.ECTRACE,
  //  DISP=SHR,LABEL=(,,,IN)
  //LINK.SYSIN DD *
  ```

```
INCLUDE USDD(TRACER,GRDSCT,IDPROF,NUPROF,CHNLIM)
INCLUDE USDD(BGNPLT,SSPPLT,BDTPLT,RNGPLT,T2DPLT,ENDPLT)
INCLUDE USDD(IDTSUB,DEEP,CONTAC,BOANG,ANGPRT,FBLOSS)
ENTRY MAIN
/*
```

- GO cards - ECTRACE uses data set reference number one for input of the depth matrix. For example, to point to a depth matrix residing in a data set called S9999.GRID on DISK03, the following DD statement is used:

```
//GO.FT01F001 DD UNIT=330,VOL=SER=DISK03,DSN=S9999.
   GRID,DISP=SHR,LABEL=(,,,IN)
```

  If a depth matrix from an outside source is used, it must be in the format specified in the Output Grid Data Format section of Appendix A. If a punched card output is not desired to make an ECCOM run, include the card

```
//GO.FT07F001 DD DUMMY
```

  ahead of the GO.FT01F001 DD statement shown above.

B. DATA CARDS

Following the

```
//GO.SYSIN DD *
```

card, the user selected data must be provided in the following format. FORTRAN field descriptors are in parentheses. The FORTRAN variable names are listed for reference.

- Card 1 (20A4)

  OTIT - title of the run

- Card 2 (9F8.2)

  X0 - source position for x coordinate (kilometers).

Y0 - source position y coordinate (kilometers).

Z0 - source depth (meters). A source depth which exceeds the water depth will be decreased to one meter above the bottom. Source coordinates in a nonpropagating location will also be adjusted.

XREC - receiver position x coordinate (kilometers).

YREC - receiver position y coordinate (kilometers).

ZREC - receiver depth (meters). The receiver may be positioned at the source if calcuations are not desired. Excessive depth is automatically corrected.

FREQ - frequency (Hz). A ray is terminated when the water depth becomes less than ten wavelengths during propagation.

FDL - (optional) segment length (meters). The parameter affects the smoothness of the vertical plot. Smaller values increase smoothness and run time. Large values create the risk of missing a bottom peak. The default value is one wavelength.

- Card 3 (6F8.2)

XMIN - initial x value of horizontal plot (kilometers).

YMIN - initial y value of horizontal plot (kilometers).

The above values are used for drawing the axis of the horizontal plot and for determining the section of the input matrix to be read into a working depth matrix (ZB). The values for KMX and KMY, recorded cell resolution (DHN), and the limits of the input matrix ultimately

determine the axis ranges. The user should be familiar with these parameters when choosing the origin of the plot.

ZST - initial depth value for vertical plot (kilometers)

ZEND - final depth value for vertical plot (kilometers)

ARST - initial range value for vertical plot (kilometers)

AREND - final range value for vertical plot (kilometers)

The above values are used to draw the axes of the vertical plot. The initial values may be negative. The program adjusts the final values if necessary for plot esthetics.

• Card 4 (6F8.2)

HDST - heading of the first ray in each ray fan (degrees)

HDEND - heading of the last ray in each ray fan (degrees)

ELST - elevation angle of the first ray fan (degrees)

ELEND - elevation angle of the last ray fan (degrees)

Headings are measured from grid north. Elevations are measured from the horizontal plane, positive downward (toward increasing depth).

HORLIM - maximum ducting or channeling range (kilometers). This is a gate value, like BLMAX CARD 2, which terminates the trace of a ray when exceeded. HORLIM serves to screen out those rays which do not appear to contact the bottom.

SMNGL - (optional) small angle value for alternative cosine difference calculation (degrees). A value of less than three degrees is recommended. See the discussion on "small angle approximations"

in Chapter III. The default value of two degrees is set when left blank.

- Card 5 (6I5)

  NHD - number of rays per ray fan (≥ 1)

  NEL - number of ray fans per bundle(≥ 1)

  NSTOP - maximum number of bottom bounces per ray (≥ 1)

The program restricts NHD to 30 rays, with a maximum total number of 100 rays produced, each with a maximum number of 50 bottom bounces or 200 reversals.

  INCPLT - number of ray segments defining one plot segment (≥ 1). With sufficiently small ray segments it is not necessary to use each iteration for plot definition to achieve a smooth plot, so this parameter is provided to enable the user to coordinate FDL and INCPLT for run time economy.

  NLEN - length of the longest horizontal plot axis (≤ 19 inches). This value controls the physical size of the plot. A value less than four will supress the plot (resulting in an ABEND code in the HASP log), but it will not affect the calculations or printed and punched output. A value greater than 15 requires additional JCL for strip plotting, since the total plot width is NLEN plus six inches. Refer to Ref. 7 for strip plotting.

NSSP - number of sound speed profiles (water masses provided as data ($1 \leq$ NSSP ≤ 3).

- Card 6 (A4,I3)

  OMAS (1) - name of first water mass

  NVAL (1) - number of (depth, sound speed) pairs defining the sound speed profile ($1 \leq$ NVAL ≤ 50).

- Card 7 - (through card 16 if necessary) (10F8.2)

  These are the data pairs (depth, sound speed), in meters an m/sec., from the sea surface to the last data point in the profile. The gradients will be calculated linearly between these points, with the final gradient below the last point set to 0.01668 m/sec. by the program.

- Next card (4F8.2)

  If the previous data is to be followed by the data for another water mass, this card must be included to define the frontal boundary between the two. The order in which the water masses appear as inptut determines their positions relative to the fronts. Cards which describe the next water mass, follow the frontal definition card and are formatted as described cards 6 and beyond. The last data card is the last string of (depth, sound speed) pairs of the last water mass; refer to Appendix M (second page) for an example of three water masses. The following variables define the boundary between each water mass:

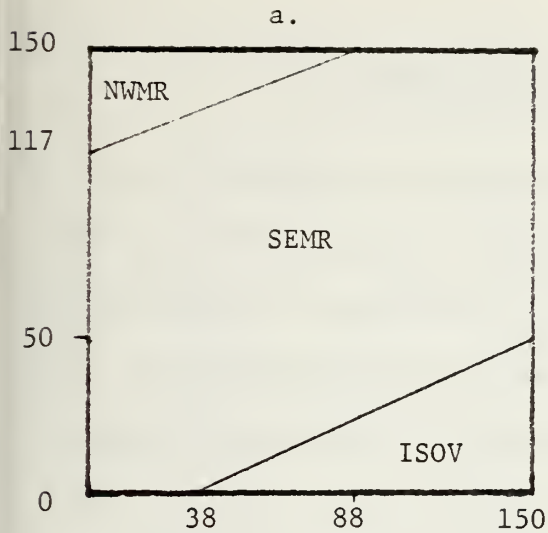
  FENTX (1) - x coordinate of front beginning point (km)

  FENTY (1) - y coordinate of front beginning point (km)

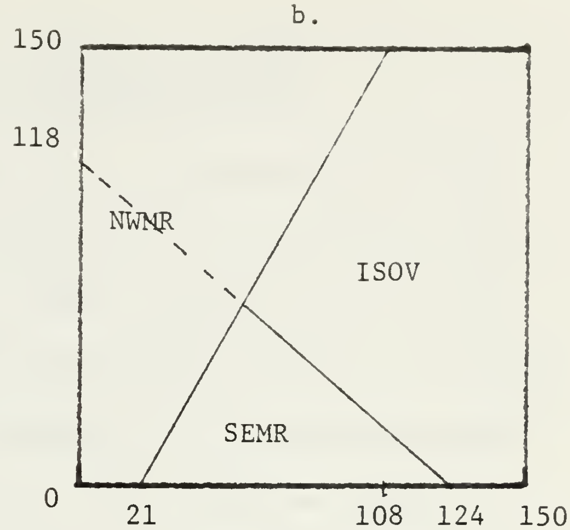
FENTX (1) - x coordinate of front end point (km)

FEXTY (1) - y coordinate of front end point (km)

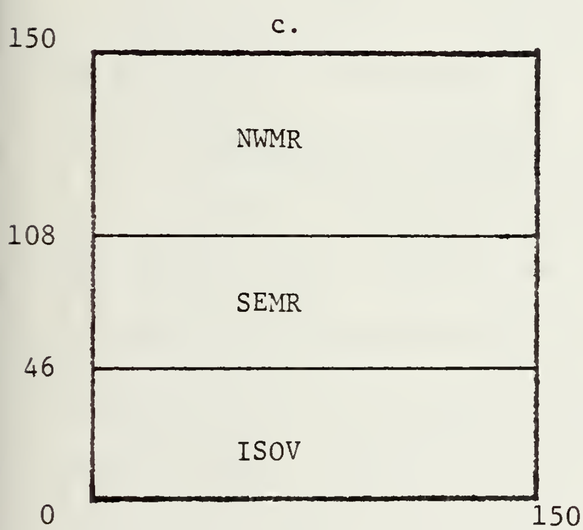
The examples shown in Fig. IV.1 identify the water mass boundaries for various input axis values in the form (FENTX, FENTY), (FENTX, FEXTY) for each water mass boundary. The first example a. represents the data from an original ECTRACE run, refer to the second page of Appendix M. Exambles b. and d. show the priority of the water masses. The first water mass defined, Northwestern Mohns Ridge (NWMR), dominates the second, Southeastern Mohns Ridge (SEMR), and the third, isovelocity (ISOV), where they overlap.



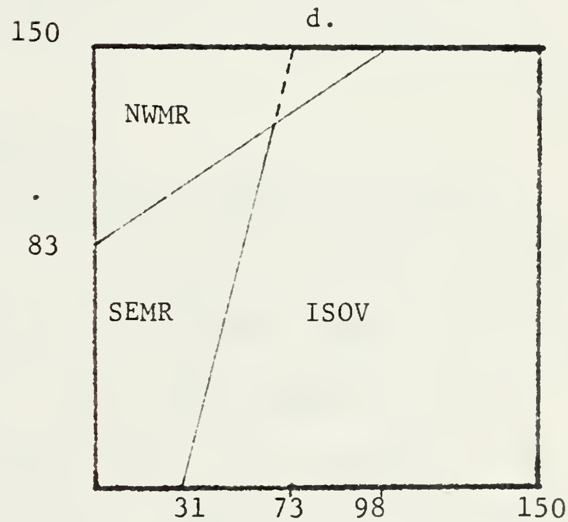
NWMR
(0,117) (88,150)
SEMR
(38,0) (150,50)
ISOV



NWMR
(21,0) (108,150)
SEMR
(0,118) (124,0)
ISOV



NWMR
(0,108) (150,108)
SEMR
(0,46) (150,46)
ISOV



NWMR
(0,83) (98,150)
SEMR
(31,0) (73,150)
ISOV

FIGURE IV.1. Examples of water mass boundaries for various values of (FENTX,FENTY) and (FEXTX,FEXTY).

V. INTERPRETATION OF ECTRACE OUTPUT

The following discussion refers to an ECTRACE run titled "Trial 67***Synthetic Basin***" (Figure 5), which is a trace conducted in a synthetic parabolic basin (Figure 6). A portion of the printed output is shown in Tables II - V. In this ECTRACE run, eight ray fans were selected to be projected between -30 and +30 degrees elevation. Each ray fan was composed of only one ray, thus the initial heading for each ray was 010 degrees.

A. PRINTED OUTPUT

Refer to Table II through V (Appendix G) as examples of the following discussion.

1. Parameter Tableau (Table II)

After processing and adjusting the input parameters just described, ECTRACE prints a final tableau of these parameters plus some additional parameters calculated from the original input such as layer gradients, the total number of rays to be traced, and the spacing between rays. In addition, the results of the depth matrix-grid load is summarized, revealing data from the header record of the grid file.

2. Ray History Tableau (Table III)

During the trace of each individual ray, this tableau prints a line of ray parameters at each end point and reversal. The following is a description of one line of parameters.

- TYPE identifies the nature of the reversal, and is

abbreviated INIT for initial position, BOTT for bottom bounce, SURF for surface reflection, OVER for turnover, and STOP for terminated position.

- X, Y, and DEPTH are the coordinates in kilometers of the reversal and end point.
- GRID HEADING is the ray heading measured clockwise from grid North at the completion of the reversal.
- ELEV is the ray's elevation angle measured downward (positive toward increasing depth) from the horizontal zero degree reference).
- BRNG TO REC is the bearing from the reversal point to the receiver position.
- DIST TO REC is the slant range distance from the reversal point to the receiver.
- WTR MASS is the water mass identifier indicating the sound speed profile applied in the present calculations.
- LYR is the constant gradient layer number corresponding to the order input as profile data (i.e., layer 1 is the surface layer and indicates that the surface gradient applies.
- DEPTH LIMITS are the physical and refractive sound channel limits of the ray, determined by its invariant. If MIN is 0.0, an upward traveling ray will reflect off the surface before reaching its upper vertex (turnover) depth. If the MAX depth is greater than or equal to 10km, a line of asterisks is printed since the ray could never attain its turnunder depth before reaching the bottom.

Otherwise, these limits are the ray's vertex depths, and if the reversal type is OVER or UNDR, the appropriate limit should exactly match the depth value on the same line.

- C is the sound speed at the reversal point. This is the ray invariant used for tracing.
- GRAZ ANGLE is the angle in degrees between the ray vector and the line of intersection of the bottom facet plane with the vertical plane. This is the angle used to calcualte bottom loss and the new ray direction vector.
- BTM ANGLE is the angle between the bottom facet plane normal and the vertical (z-axis), in degrees.
- BTM LOSS is the accumulated intensity loss in dB after the bottom reflection. When this value equals or exceeds BLMAX, the ray's trace will terminate.
- LPS is the number of steps required by the CONTAC subroutine to locate the point of bottom contact. A long ray segment length may require more steps than a shorter one. A maximum number of 50 steps is permitted before the bottom facet is assumed anomalous and the ray's trace is terminated.
- Prior to the final line of an individual ray's trace history, is a summary of plot definition parameters plus data on the ray's CPA to the receiver, and reason for ray trace termination. The CPA calcualtions are accurate to within one ray segment length.

3. Ray Fan Summary Tableau (Table IV)

At the completion of the trace of one complete ray fan, a table is printed listing each ray's elevation angle and depth in meters before each bottom bounce. All zero values at the bottom of each ray's column indicate blank data.

4. CPA Summary Tableau (Table V)

When all ray traces have been completed, a table is printed listing each ray's heading, time, and distance at CPA to the receiver.

B. PLOTTED OUTPUT

ECTRACE produces a single montage of up to five separate plots (Fig. 13), the total dimensions of which depend upon the user's selection for the maximum size in inches of the longest axis of the horizontal plot (NLEN). The following table lists some sample values for NLEN and the resulting dimensions of the complete plot.

NLEN	(in.) Maximum Height	Maximum Width	Remarks
5	7	8.33	minimum
10	14	16.67	
15	21	25	maximum size without strip plotting [5]
19	26.6	31.67	maximum allowable

Components of the ECTRACE plot follow.

1. Sound Speed Profiles (Figure 13)

A plot of sound speed versus depth is made for each water mass defined.

2. Vertical Plot (Figs. 15 and 16)

Most two-dimensional ray trace programs produce plots of ray paths on graphs of depth versus range. The vertical plot is similar except that all ray paths are projected from three dimensions onto the vertical plane. The plane is oriented along the mean ray fan heading and includes its profile of the sea bed. In ECTRACE trial 65B (Figure 13) the ray fans each contain two rays whose initial headings vary from 360 to 020 degrees as indicated by 10.000 +/- 10.000 on the plot. The bathymetry profile is made along the heading of 010 degrees. On a plot, the heading value (HEADINGS) always indicates the direction of the bathymetry slice. The (+/-) value signifies the initial heading of the first and last rays and that additional rays are evenly spaced in the interval.

Since the rays are actually traced in three dimensions, bottom reflections occurring outside the vertical plot plane may appear to be taking place above or below the bottom surface instead of at the boundary (Fig. 15). Only those bottom contact points which occur in the projection plane must lie on the seabed profile.

The symbols listed on the plot (Fig. 5) are used to identify bottom contact locations in the vertical and horizontal plot, and are unique for each ray fan except that up to thirteen symbols are used before repetition, as shown in Table VI.

3. Horizontal Plot (Fig. 17)

This plot is the most important plot product of ECTRACE and is the plot of the ray paths on a horizontal plane

(topview). This plot displays the X-Y positions of each bottom contact point using the ray's (ray fan) identification symbol and connects them with straight lines, revealing the horizontal deflection effect. Each ray's plot begins at the source, ends up at the last bottom contact point in the grid and is labeled with the ray number (Fig. 17). The ray number can be used to locate its printed and punched output history.

The axes are labeled with respect to the origin of the input matrix. The geographic center position and coordinates of the input matrix, if included in the bounds of the working matrix (FORTRAN array ZB), is also plotted for reference. It is important to note that this plot, without the bottom contact symbols, may combine the results of other ECTRACE runs by combining the optional punched card outputs as input to ECCOM (Figs. 7 and 18).

C. PUNCHED OUTPUT

The punched card output for a single ECTRACE job consists of the following cards:

CARD	DATA CONTENTS
1	The runtitle
2-3	The grid title
4	Grid reference coordinates and scale factor
5	Initial elevation, initial heading, number of points (BB) and ray number of the first ray
6+	X-Y coordinate pairs of plot points of the first ray
(Data for additional rays is repeated as 5 and 6+ above)	
Last	9999., to flag the end of data

These cards may be stacked as a member with those of several ECTRACE jobs as input to a single ECCOM job. For ECCOM to be meaningful, cards 2 through 4 of each stack member should be identical, and the user should verify this himself.

Punched card output may be quite voluminous and should not be produced unless required. The following card, inserted before the first GO card (JCL), will suppress the punched output:

```
//GO.FT07F001 DD DUMMY      .
```

VI. SUMMARY

A. POTENTIAL APPLICATIONS

ECTRACE, in its present form, has many potential applications:

- The investigation of asympotic deflection angles in ocean areas where bathymetry data are well-documented, allows optimum hydrophone positioning. For example, Ref. 6 cites cases where seemingly attractive receiver positions would be affected by shadowing in the horizontal plane.
- Adaptation of ECTRACE ray tracing methods could be incorporated into existing programs which presently rely upon an assumption of radial symmetry of ray propagation about a source or receiver (no horizontal deflection).
- The horizontal deflection effect may lend additional insight into the study of travel times and intensities of acoustic signals, enabling more accurate source fixing information.

B. IMPROVEMENT

There are many areas of improvement which would increase the utility of ECTRACE and its supporting computer programs for the above applications:

- A better routine to convert bathymetry contour data into a matrix of depth values is essential for modeling

regions for which only contour data exists. Most known interpolation algorithms perform best on data which are randomly or regularly spaced rather than in the form of contours. For example RAIN2 [5] would probably perform better with the data used to make the contours than it does with the contours themselves. While the contour data are useful for cartographic and geologic purposes, bathymetry data in a less processed form are significantly better sources for constructing computer models of the sea bed.

- ECTRACE in its present form makes no calculations of intensities or their losses due to spreading, attenuation, or scattering, thus preventing its use as a propagation loss model.
- An improved bottom loss function is recommended over the supplied FORTRAN function subprogram FBLOSS. The user may perform this substitution easily by deleting FBLOSS from the INCLUDE step in the LINK statement of the ECTRACE job control and supplying another function subprogram (added at the end of the calling program) that accepts a double-precision argument for grazing angle and is also called FBLOSS.
- The assumption of specular reflection neglects the phenomenon of transmission into the sediment and its effect on the location of the reflected ray due to sub-bottom boundaries. It would be interesting to incorporate in an improved version of ECTRACE.

APPENDIX A
GENBOT USER INSTRUCTIONS AND OUTPUT DESCRIPTION

GENBOT is used in the construction of a depth grid file from bathymetry contour files. In this discussion, the term "grid" refers to a matrix of depth values whose row and column dimensions represent data point spacing in terms of a desired resolution parameter DHN.

1. Selection of Bathymetry Grid Regions

GENBOT will create a bathymetry grid file for any region of the geoid for which contour data is supplied in the format stated in section A-4. The user must select the latitude and longitude of the center of the region of interest, a radius in kilometers and the cell resolution in kilometers. GENBOT will attempt to create a square matrix whose semi-axes represent the selected radius and whose elements represent depth values in kilometers.

A 30-km data margin, outside the calculated axis limits, is required to begin interpolation. Should this requirement not be met, GENBOT will symmetrically reduce the length of one or both axes until the resulting rectangular matrix plus margins with the desired center position intact is totally encompassed within the data region. The center coordinates, area radius, and cell resolution must be selected so as to avoid generating a grid which is impracticably small. In particular, a generated grid may not overlap a pole, and one with a pole on a margin boundary must be supplied data from eighteen contour files to cover 180 degrees of longitude.

Contour data currently available in usable format cover 40 degrees of longitude of the North Atlantic Ocean, specifically the Norwegian and Greenland Seas, from longitudes 20W to 20E and latitudes 60N to 90N (Fig. 8). The Naval Research Laboratory, Acoustics Division, Arctic Section,

has additional files which cover the entire North Atlantic land and sea regions which may be utilized by GENBOT after modification. However, GENBOT need not be the exclusive means of grid instruction. The only requirement is that the grid exists as a file in the format specified. The user may wish to draw on other souces of bathymetry data, such as SYNBAPS [8], and tailor other available resources to construct a usable grid.

2. Grid Construction Techniques

Once supplied with the user's selection for the grid title, center coordinates, cell resolution and area radius (TTL, CNLAT, CNLON, DHN, ARAD), GENBOT begins by defining a north-oriented square region whose boundaries are at a distance ARAD from the center. The subscripts of the two dimension matrix depicting this region will represent equal units of DHN spacing from the origin along the x and y axes of the square using a cartesian coordinate system. Thus, while all distances calculated during grid generations are geodetic distances from the center, point positions are made relative to the grid origin or southwest corner. This technique results in a plane projection which is distortion free along straight tracks which pass through the center. Subroutine CORNER then calculates and prints a table of the initial cartesian and geographic coordinates of the grid corners. The cartesian coordinates will always range from (0,0) to their maxium values, initially double the value of ARAD. The geographic coordinates should reflect some meridian convergency. This table will be repeated for the final grid, showing the adjustments made, if any, to meet the data requirements.

Many geographic coordinate pairs in an input file may be well outside the area relevant to the grid, so GENBOT calculates gate values called

SLAT, ELAT, SLON, and ELON for data discrimination. SLAT and ELAT are always one degree beyond the minimum and maximum latitude values calculated by CORNER, but SLON and ELON may extend beyond the minimum and maximum longitudes by 1, 10, or 20 degrees, depending upon the maximum latitude.

The grid region and its data environs recognized by GENBOT at this stage is shown in Fig. A.1. All points within these environs are subject for consideration during interpolation of depth values in the grid.

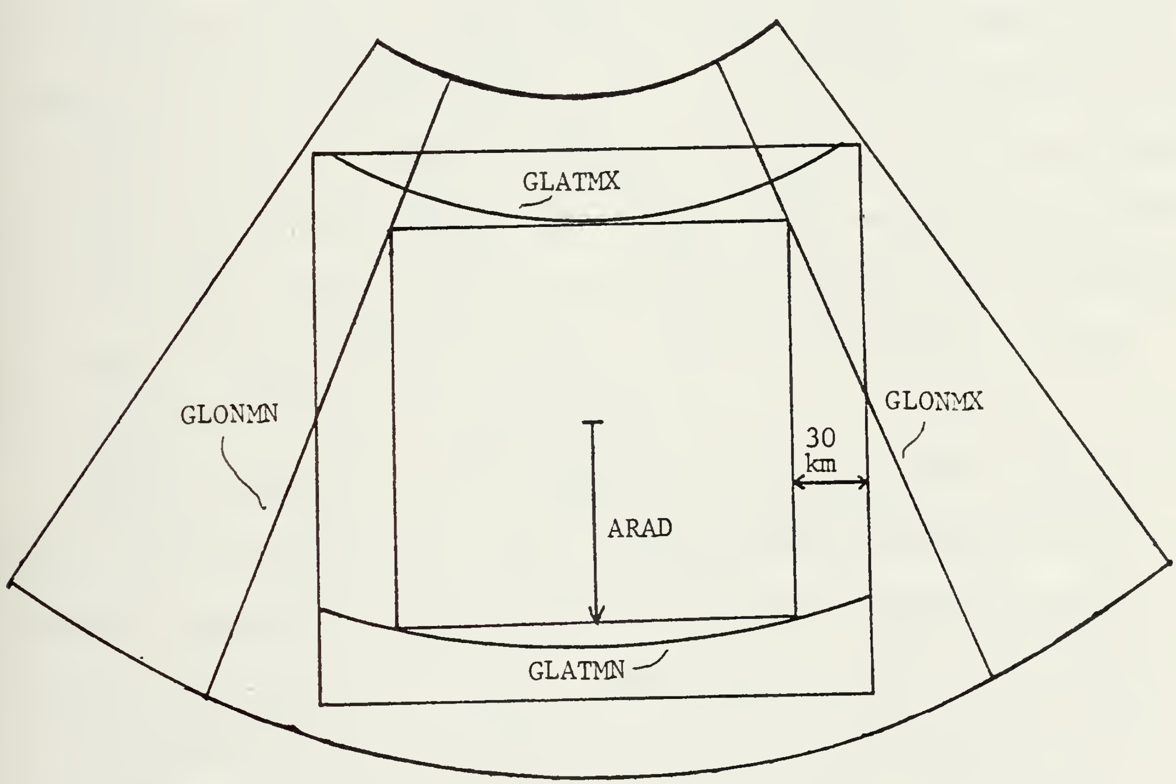


FIGURE A.1. Geographic region used for grid construction.

Data files used as input for GENBOT are digitized represenations of an NRL bathymetry chart [4] and were recorded on Mobil Oil Company Tape NAV-78 for use in a computer program to draw portions of the chart in the form of bathymetric contour lines and their depth labels as they exist on the chart. The records are in the card image format and consist of three types:

- NAME - depth value coordinates
- CON - contour line header label
- Chart coordinates of points along the contour line

In the original form the numerical data on the tape consisted of latitude/longitude pairs, providing no means of determining the depth represented by the contour lines without visual inspection of the computer plot. The tape used in this research is similar in format to NAV-78 except that the contour line header records include depth data (provided by NRL), and only the regions shown in Fig. A.2 and listed in Table A.I, are covered. The numerals of the data set names were chosen to reflect the original file number on the source tape, NAV-78.

Using the magnetic tape files selected by the user, GENBOT begins reading the digitized contour data, stopping after each record to discriminate the points, calculate geodetic distances, transform the data to x, y, z coordinates, and load the coordinates into three axis vectors, AX, AY, and AZ (which will be used later for interpolation) while simultaneously keeping a running plot of the contour lines.

After all input files have been processed, the vectors AX and AY are sorted into ascending order by the SSP3 subroutine RAIN1 [5]. The minimum and maximum values are tested to determine if the 30-km outside data margin is included in the range. If not, axis reduction begins until

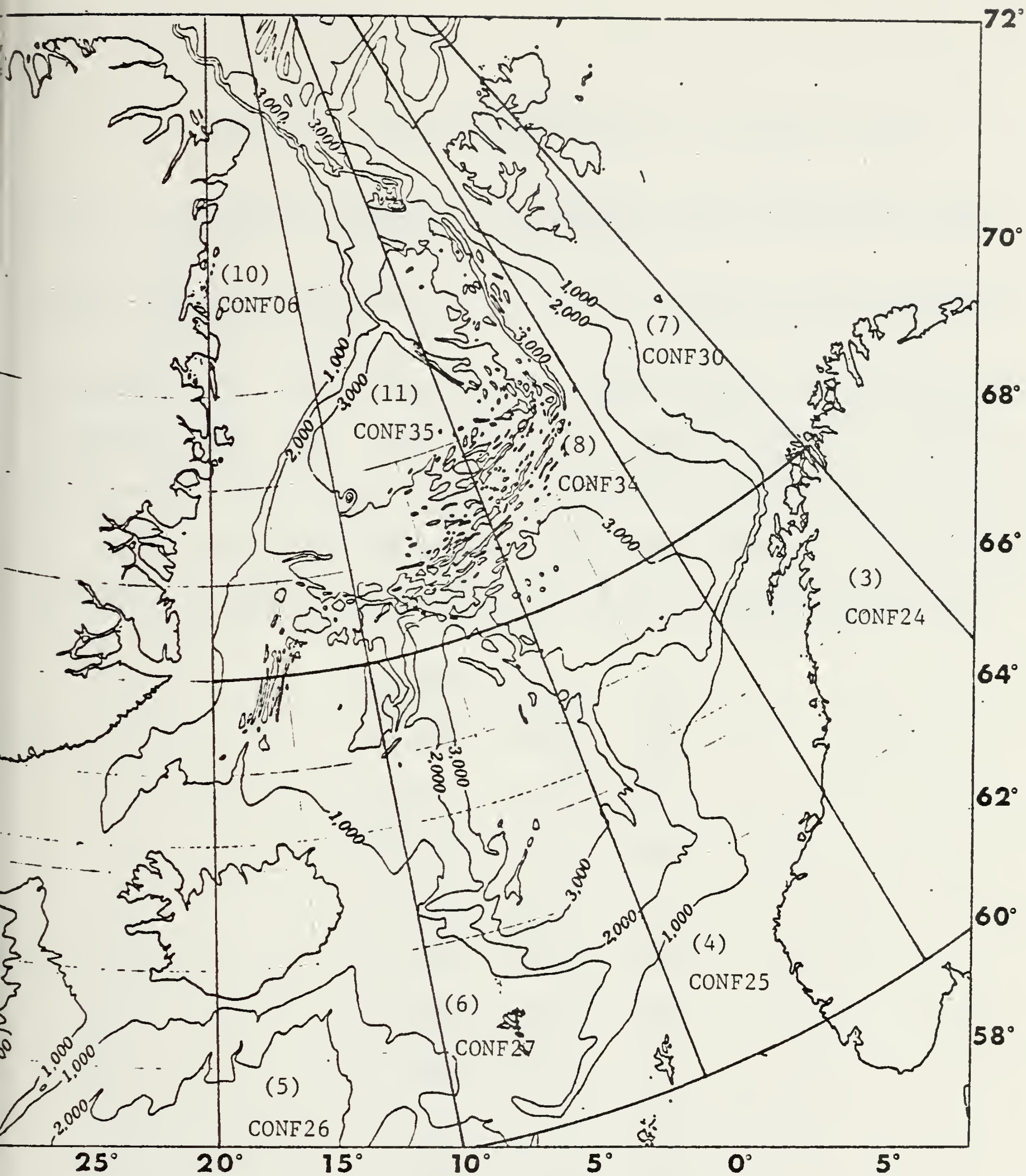


FIGURE A.2. North Atlantic regions, with their magnetic tape file numbers and data set names, available for use by GENBOT. All are 10° by 10°. Other regions available on the same tape are listed in Table A.I.

TABLE A.I
FILE ONE OF CONTOUR DATA RECORDED ON A MAGNETIC TAPE

DATA: BATHYMETRIC CONTOURS OF THE NORWEGIAN AND GREENLAND SEAS FROM LAT 60N TO 90N and LONG 20W to 20E.

SOURCE: ADAPTED FROM MOBILE OIL CO. TAPE NO. NAV-78 AND NAVAL RESEARCH LABORATORY DATA.

STRUCTURE: EACH FILE COVERS A 10 X 10 DEGREE REGION. THE FOLLOWING LIST IDENTIFIES THESE FILES. THE FIRST VALUE IS THE FILE NUMBER, FOLLOWED BY THE DSN, THEN THE LAT/LONG PAIR OF THE SOUTHWEST CORNER (LABOR,LONOR) OF THE REGION, AND FINALLY THE NUMBER OF CARD IMAGES CONTAINED IN THE FILE. THE TAPE IS STANDARD LABELED, DENSITY 1600 BPI, LRECL=80,BLKSIZE=800.

2	CONF13	+80-20	256
5	CONF26	+80-20	4343
8	CONF34	+70+00	12443
11	CONF35	+70-10	9428
3	CONF24	+60+10	569
6	CONF27	+60-10	6001
9	CONF37	+80+00	491
12	CONF36	+80+10	409
4	CONF25	+60+00	3026
7	CONF30	+70+10	857
10	CONF06	+70-20	4454
13	CONF38	+80-10	972

either the margin requirement is met or goes below a 7.5 km semi-axis length, the latter case preventing grid construction although a contour plot may still be produced. Subroutine CORNER is then called again to print the corner coordinate table. If no adjustments were necessary, the second table will be identical to the first.

The interpolation process begins at the southwest corner of the grid and proceeds in a columnar direction (i.e., completing the range of x_i values for each y_j value) via successive calls to the SSP3 subroutine RAIN2. The accuracy of the interpolated values is proportional to the density of data in the given area of interest. Every ten values interpolated are written as a record on the output file. The process continues until the total number of matrix elements has been reached, at which time the output file is closed and the program is terminated.

3. User Instructions

a. JCL and Program Deck.

The program section of a sample deck is printed below. The TIME and REGION.GO parameters were chosen in anticipation of a maximum number of 5000 data points in the area of interest, a square region of 210 km on a side (150 km plus margin). The program listings included in this appendix are part of this example.

```
//(Standard job card),TIME=20
// EXEC FORTCLGW,REGION.GO=250K (See Appendix O.)
// FORT.SYSIN DD *
   (GENBOT, CORNER, GEOPLT, GEODST listings)
   (RAIN1, RAIN3, RAIN2 listings)
/*
```

The following DD statements identify the magnetic tape files representing the ten by ten degree regions into which the subroutine

extends. In the example, the area chosen is centered at coordinates 70°-04'N/15°-51' (70.07059/15.85302) with a radius of 75 km. This area requires two data files as input, CONF24 (file 3) and CONF30 (file 7). Since tape data is sequential, the order of the DD cards should be in the order of the file numbers (appearing in the LABEL parameter).

```
//GO.FT01F001 DD UNIT=3400-3, VOL=SER=NPS,DISP=(OLD,PASS),
//   LABEL=(3,SL,,IN),DSN=CONF30
//GO.FT01F002 UNIT=AFF=FT01F001,VOL=(,RETAIN,REF=*.GO.FT01F001),
//   DISP=(OLD,PASS),LABEL=(7,SL,,IN),DSN=CONF30
```

More information on magnetic tape usage at NPS may be found in Ref. 10.

Next must follow the DD statements specifying the data set and device for permanent storage of the generated grid. The permanent device may be another magnetic tape, a disk, or punched cards. In all cases, the size of the output data set is important, and the DCB parameters should be considered in advance if the grid size is different from the example. GENBOT produces card image out put, meaning the logical record length is 80 bytes.

- Punched cards. If the grid is to be stored on punched cards, the user need only change the data set reference number on the appropriate WRITE statements in GENBOT from a two to a seven. The number of cards punched will equal a 3+[(row dimension)(column dimension)/10] rounded up to an integer value. This represents three header cards followed by ten depth values per card in columnar order, as required by ECTRACE. Thus if the input grid radius was 75 km and input cell resolution 1 km, the maximum parameter calculated by GENBOT will be:

XMAX	YMAX	IMAX	JMAX
150	150	151	151

resulting in a matrix of 22801 elements and a punched output of 2284 cards.

- Disk. Grid storage on a disk device is the most convenient form for multiple ECTRACE runs over the same sea bed. The following DD statement stores the example grid on a 3330 disk drive device at NPS as a data set named S999.GRID.

```
//GO.FT02F001 DD UNIT=3330,VOL=SER=DISK03,DISP=(,KEEP),
//   DCB=(FECFM=FB,LRECL=80,BLKSIZE=800),SPACE=(TRK,20),
//   DSN=S9999.GRID
```

Blocked at 800 bytes, the output data requires 14 tracks per 10 records (card images). More information on the DCB and DSN parameters may be found in Ref. 11. When the generated grid is no longer needed, and if the user wishes to generate a new grid, or if space has already been allocated on the device, the DISP parameter should be SHR in place of (KEEP) and the DCB and SPACE specifications may be omitted. Disk data sets are periodically purged by center operations personnel, but obsolete data set space should be cleared or reallocated by the user.

b. Input Data.

The following input data are user selected values and follow the //GO.SYSIN DD* card.

Card	Variable	FORTRAN Field Descriptors	Remarks
1-2	TTL	6A8/6A8	Title
3	CNLAT, CNLON, DHN, ARAD	4F10.5	center latitude, center longitude, cell resolution, radius
4	NFILES, LEN	2I5	Number of input files, plot length

The magnitude of LEN determines the plot size ($\leq$19 inches). If LEN is negative, no grid will be constructed. If its magnitude is less than four, no plot will be produced.

c. Printed Output

GENBOT printed output supplies the following:

- A repeat of the user input data.
- A table of the grid corner coordinates and the depth matrix dimensions using the input parameters.
- A summary of results for each contour line, including its depth, and the title from the header label, the number of points on the line (NPTS) and the number of points used from the line (KPTS).
- A corner coordinate table of the final grid.
- A summary of the interpolation results. A line is printed for each interpolated point whose accuracy may be questioned. The meaning of the IER parameter may be found in the RAIN1 listing in Appendix I.

d. Plotted Output

The contour lines of the input file are plotted with the local parallels and meridians overlaid. The resultant subregion represented by the output grid is outlined by a rectangle with labeled axes. The axis values are referenced to the desired grid origin and not the resultant grid origin, the latter being at (0,0) always.

e. Programming Notes on Contour Data Interpolation

Subroutine RAIN2, the interpolation routine used by GENBOT, is aptly suited to interpolate points that are more or less randomly spread in both horizontal directions. The nature of contour data, however, is typically a collection of closely spaced depth values along widely spaced

contour (compare Figures III.1. and A.1.). This type of data arrangement frequently results in a colinearity problem for RAIN2, which can be resolved only by searching ever widening data cells for a non-colinear point, resulting in a degradation of accuracy. In addition, the routine is highly sensitive to contour line kinks which result in right-angle deviations in the interpolated contour lines (which may be plotted by GRDCHK).

Correcting the deficiencies of an interpolating routine is beyond the scope of this research, but the Naval Research Laboratory, Arctic Section, is currently investigating methods for generating grid-matrices from contour data. Users are encouraged to modify the code used in GENBOT as desired to produce results which more closely reflect the actual data supplied.

4. Input Contour Data Format

As stated previously, some contour data already exists in a format usable by GENBOT, but if data is supplied by another source, it must be arranged in a file in the manner and format specified below. All records are 80 byte card images.

The first record of the file contains data on the file itself. GENBOT uses a 4I5 field to represent the origin latitude/longitude pair, the number of contour lines in the file, and the number of card images it may expect to read.

Contour line data exists as separate sequences of cards and may appear in any part of the file as long as each line or line segment is in an unbroken sequence with a header card. The header card must contain the line's depth in hundreds of meters, right justified in a I4 field in columns 41-44. The cards following the header card contain the latitude/longitude pairs in degrees and decimals of degrees of the points along

the contour line in the form of four pairs of coordinates (in FORTRAN fields 8F10.5) per card. South and west values are negative. A value greater than 90.0 in magnitude in any latitude field is interpreted as the end of data for the contour line, and GENBOT expects the next record to be either another contour line header label, the end of the file, or a depth value position record.

Depth value position records are optional and are used during the chart-drawing phase of the program to place alphanumeric symbols on the chart product. They may exist any place after the first record in the file except imbedded in a contour line sequence. Columns 4-8 must contain "NAME" in an A4 field, columns 19-38 contain the latitude and longitude of the symbols in a 2F10.5 field and the symbols themselves must be in columns 46-47. These records are normally used to draw depth values in hundreds of meters on the contour lines.

If the user wishes to use GENBOT on externally supplied contour data, it it recommended that a dump be performed on one of the smaller files (such as CONF13) of the GENBOT data tape to verify the required format for adaptation.

5. Output Grid Data Format

GENBOT constructs the output file in accordance with GRDSCT requirements. Since GRDSCT is the subroutine used by TRACER (of ECTRACE), G3DP and GRDCHK, any externally supplied grid data must all be of the same structure as outlined below in order to be used by those programs. Again, all records are card images.

a. The first two records contain the input grid title in the 10A8 fields.

b. The third record contains the input latitude and lontitude of the

grid center, the point spacing, and the number of rows and columns of the matrix representing the grid. The record format is (3F10.5,2I5).

c. The fourth and remaining records contain ten depth values per card, in kilometers, as 10F8.4 fields, beginning with column one (y = 0) and proceeding all x-values before going to the next column, in accordance with the FORTRAN convention of columnar storage of matrices.

APPENDIX B
SYNGEN DESCRIPTION AND USER INSTRUCTIONS

1. Program Description

a. SYNGEN is a collection of simple functions which may be selectively used to generate and record a matrix of depth values representing a desired synthetic bottom configuration. Six bottom cases are provided (see below), but the program may be easily modified by the user to produce a case not described here. A program listing is provided in Appendix H.

The user selected parameters are:

- CNLAT - any desired center latitude value.
- CNLON - any desired center longitude value. These parameters are requested in compliance with the required output file format.
- DMIN - the minimum depth value (D_1) to be attained within a 75 km radius from the grid center.
- DMAX - the maximum depth value (D_0) within the same radius.
- ITYPE - an integer code for the bottom type desired.

b. Table B.1 lists the ITYPE code, the bottom type, the functions used to generate the depth matrix, and the matrix dimensions. All grids represent a region covering a 150 km in both the x and y directions, but the matrix size will be 11 by 11 or 151 by 151, depending upon the bottom type (DHN values are 15 km and 1 km, respectively). In all cases, $\Delta D = D_1 - D_0$.

c. The depth file provided by SYNGEN conforms to the format required by ECTRACE, GRDCHK, and G3DP. When one of these programs uses a SYNGEN product, the array dimension in the program's source code must be checked to conform to the dimensions of the generated matrix.

TABLE B.I
SYNTHETIC SEA BED FUNCTIONS

ITYPE	Type	Function	Parameters	Matrix Dimensions
12	Wedge	$H = D_1 - \gamma$	$\gamma = \Delta D/135$ $0 \leq x \leq 135$	11 x 11
13	Ridge	$H = D_0 + \lvert\gamma\delta(x)\rvert$ $H = D_0$ elsewhere	$\gamma = \Delta D/15$ $\delta(x) = x - 75$ $60 \leq x \leq 90$	11 x 11
14	Trough	$H = D_1 - \lvert\gamma\delta(x)\rvert$ $H = D_1$ elsewhwere	$\delta = \Delta D/15$ $\delta(x) = x - 75$ $60 \leq x \leq 90$	11 X 11
15	Conical Seamount	$H = D_0 + \gamma r$ $H = D_0$ elsewhere	$\gamma = \Delta D/75$ $r = \sqrt{(x - 75)^2 + (y - 75)^2}$	151 X 151
16	Sinusoidal Undulation	$H = D_0 \gamma[1+\cos\{\delta(x)/R\}]$	$\gamma = \Delta D/2$ $\delta(x) = 2(x - 75)$ $R = 15$	151 X 151
17	Parabolic basin	$H = D_1 - \gamma r^2$	$\gamma = \Delta D/(75)^2$ $r = \sqrt{(x - 75)^2 + (y - 75)^2}$	151 X 151

2. <u>User Instructions</u>

THE JCL of a sample deck are arranged as follows:

```
//(Standard JOB card),TIME=2
// EXEC FORTCLG, REGION.GO=250K
//FORT.SYSIN DD *
  (SYNGEN source code)
/*
//GO.FT02F001 DD UNIT=3330,VOL=SER=DISK03,
// DSN=S9999.SYNBED, DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),
// DSP=(NEW,KEEP), SPACE=(TRK,20)
//GO.SYSIN DD *
  (user selected data)
/*
```

Notice that the output data set is directed to DISK03. The DSN is the user's selection for the data set name in accordance with NPS system requirements. A 151 x 151 depth matrix requires twenty tracks of disk space on a 3330 device.

Only one data card is needed to supply the program with the user's parameter selections. Its field descriptor is (4F10.5,I5) and contains the values reflected in the same order as listed in paragraph 1.a.

APPENDIX C
G3DP DESCRIPTION AND USER INSTRUCTIONS

1. Program Description

a. G3DP uses the NPS library subroutine PLT3D1 [Ref. 12] to draw a perspective surface plot of a subset of a depth matrix. This plot is a normal projection of the surface using a focal length and line-of-sight calculated from the user's selections for the viewpoint position angles DEGUP and DEGCW. As the variable names imply, the viewer is elevated DEGUP degrees from a reference depth and rotated DEGCW clockwise from a south-to-north viewing aspect. For example, the standard working matrix (dimensioned 151 by 151) is best viewed with DEGUP and DEGCW values of 30 degrees and 45 degrees respectively, providing the same viewing aspect demonstrated in all the the G3DP products printed in this report. Using a DEGUP value of zero results in an isometric rather than a perspective projection. The DEGCW selection should be between zero and 89 degrees (south-north to west-east viewing aspects).

b. As in ECTRACE and GRDCHK, G3DP requires source code modification by the user to tailor the program to the desired dimensions of the working matrix. Instructions in this regard are included as comments in the FORTRAN listing (Appendix K).

c. An important characteristic of G3DP is that it uses only the odd-numbered rows and columns of the working matrix to produce the surface plot. This reduction in resolution is required to reduce core requirements and unclutter the graphic output caused by typically large (greater than 75 by 75) working matrices.

2. User Instructions

The JCL of a sample program deck are arranged as follows:

```
//(Standard JOB card),TIME=2
//EXEC FORTCLGW,REGION.GO=300K  (Appendix O.)
//FORT.SYSIN DD*
     (G3DP source code)
/*
//LINK.USDD UNIT=3330,VOL=SER=DISK01,DSN=1570. ECTRACE,
//  DISP=SHR,LABEL=(, , ,IN)
//LINK.SYSIN DD*
    INCLUDE USDD(GRDSCT)
    ENTRY MAIN
/*
//GO.FT01F


//GO.SYSIN DD*
    (user-selected data)
/*
```

The LINK step provides access to GRDSCT for loading the working matrix from the input file. The input data set is the same example used in ECTRACE and GRDCHK. The card input data are user selected and contained on one card, as follows (FORTRAN field descriptor 5F8.3)

- XST - working grid matrix x-origin relative to that of the input grid (km).
- YST - working grid-matrix y-origin relative to that of the input grid (km).
- DEGUP - viewpoint elevation angle (0 ≤ DEGUP ≤ 89 degrees).

- DEGCW - viewpoint rotation angle from south to north ($0 \leq DEGCW \leq 89$ degrees).
- LW - width of the plot ($5 \leq LW \leq 18$ inches).

APPENDIX D
GRDCHK-USER INSTRUCTIONS AND OUTPUT DESCRIPTION

1. Purpose and Output Description

Cases may arise where the accuracy of depth values in a generated matrix may be in question because of data sparsity, interpolation deficiencies or other reasons. Program GRDCHK assists in evaluating the generated grid by printing and plotting selected columns of the matrix for comparison with that of the source contours. The contour plot may also be used on a light table with an ECTRACE or ECCOM horizontal-ray trace plot (Figs. 5, 6, and 7).

The NPS library routine CONTUR is called for producing the contour plot. Subroutine GEOPLR is a modified version of GEOPLT to accommodate the rotated plot produced by CONTUR..

2. User Instructions

To aid in this discussion refer to a computer listing of GRDCHK code in Appendix J. A listing of the CONTUR subroutine may be found in Ref. 6.

The following example applies to a standard 151 by 151 working grid. Below is a listing of the JCL cards and the position of the program deck.

```
//(Standard job card), TIME=5
// EXEC FORTCGLW,REGION.GO=350K
//FORT.SYSIN DD *
  (GRDCHK,CORNER,GEOPLT,GEODST source decks)
/*
```

The following DD statements link to a member of ECTRACE for allocating a working subregion of the input.

```
//LINK.USDD DD UNIT=3330,VOL=SER=DISK01,
```

```
// DSN=1570.ECTRACE,DISP=SHR,LABEL=(,,,IN)
//LINK.SYSIN DD *
  INCLUDE USDD (GRDSCT)
  ENTRY MAIN
/*
```

Next follows the DD statement identifying the depth matrix input file. The same example is used for the GENBOT output file.

```
//GO.FTOF001 DD DSN=S9999.GRID,UNIT=3330,VOL=SER=DISK03,
//   DISP=SHR, LABEL=(,,,IN)
```

Finally the user input parameters are included. The card

```
//GO.SYSIN DD *
```

is followed by:

Card	Parameter Names	FORTRAN field description
1	NL, LEN	2I5
2	ZST, ZEND, XMIN, YMIN	4F6.1
3	IPS, IPE, JPS, JPE	4I5

where

- NL is the number of contour levels desired. The shallowest and deepest levels, if known, should be read into ZST and ZEND. If the boundary levels are not known, NL should be made negative, and subroutine CONTUR will calculate the boundary and contour levels. If NL is greater in magnitude than 100, no contour plot will be produced.
- LEN is the length of the contour plot's longest side in inches. If LEN is negative, a cartesian grid will be drawn on the contour plot instead of geographic coordinate lines. If LEN is less than 4, no contour plot will be produced. The maximum

magnitude of LEN is 19 inches.

- ZST AND ZEND are estimates of the smallest and largest values in the working matrix (ZB).
- XMIN and YMIN establish the origin of the working matrix relative to that of the input matrix. The program assignments for KMX and KMY, the recorded cell spacing DHN, and the dimensions of the working matrix determine the subregion size parameters in the same manner as they do in ECTRACE, since subroutine GRDSCT is linked to this program. In effect, the user must tailor certain portions of the program cards to compensate for the FORTRAN G inability to perform object time dimensioning.
- IPS, IPE, JPS and JPE identify the beginning and end rows and columns of the working depth matrix to be singled out for inspection. Thus, if ZB is dimensioned (150,150) and the above values are 1, 150, 74, and 78 respectively, GRDCHK will print the depth values of these 750 elements of the ZB matrix and plot five columnar profiles (Fig. 4). If this record is omitted, only the contour plot will be produced.

APPENDIX E

ECCOM OUTPUT DESCRIPTION AND USER INSTRUCTIONS

1. Description of Output

ECCOM allows the user to combine several selected ECTRACE runs and eliminate specific rays as desired to produce an improved horizontal ray path plot for investigation of curvature (Fig. 7 and Fig. 18). The size of the ECCOM plot will be the same size as first ECTRACE run of the input card deck unless the user opts to alter the scale factor (FACT) to control the plot dimensions.

The printed output is a list of all the rays processed for the ECCOM plot, giving each ray's initial elevation angle and heading. Additionally, the number of points used to plot each ray path is printed. This number of points corresponds to the number of reflections plus the starting point of the ray. Refer to Table VII for a sample listing of data for the ECCOM 51 plot, shown in Fig. 7.

2. User Instructions

The following JCL cards and parameters are required for an ECCOM run:

- JOB card - The CPU time parameter should allow four seconds for every 55 rays to be plotted.
- EXEC card - This card must specify FORTCLGW procedure.
- Calling program - Following the

  //FORT.SYSIN DD *

  is the ECCOM program deck. See Appendix L for the program listing.
- Data cards--Following the

  //GO.SYSIN DD *

  is the sequence of punched output decks, each individually arranged as described in Chapter V, Section C. The user may change the title

card (card # 1) of the first ECTRACE punched deck to identify the ECCOM run title. In addition, the user may change the scale factor (FACT) on card number 4 to modify the plot size. FACT was calculated by ECTRACE in response to the user's selection for the horizontal plot length (Chapter V, Section B). The following are sample values of FACT and recommended limits:

FACT	Horizontal Plot Axis Length (inches)
0.333	5 (minimum recommended)
1.000	15
1.267	19 (maximum recommended)

If the user desires a plot size outside the range recommended, refer to Ref. 7.

Cards 1 through 4 of each ECTRACE deck after the first are ignored by ECCOM, but the user should verify that they all reference the same portion of the same input grid-matrix in order for the ECCOM plot to be meaningful.

APPENDIX F
ANALYTICAL VERIFICATION OF ECTRACE HORIZONTAL RAY PATHS

The accumulation of ray heading changes after repeated specular reflection from a sloping sea bed describes a faceted horizontal path whose apparent curvature may in some instances be analytically described. Harrision [2] derived some solutions for horizontal ray paths over simple bottom topography cases. Most of these cases may be modeled by SYNGEN and used to test the validity of the horizontal ray paths mapped by ECTRACE. We chose an isovelocity sound speed condition and a conical seamount because its surface geometry effectively challenges the facet method of topographical modeling.

For a conical seamount of arbitrary slope but whose apex just touches the sea surface, Harrison gives the ray paths in polar coordinates (r, Φ) as

$$r = r_0 (1 - \cos^2\theta_0 \cos^2\phi_0)^{\frac{1}{2}}$$

$$\cdot \operatorname{cosec} \left| \frac{(1 - \cos^2\theta_0 \cos^2\phi)^{\frac{1}{2}}}{\sin\phi_0 \cos\theta_0} (\Phi + \Phi') \right| , \qquad \text{(F.1)}$$

where
r_0 = initial distance from the apex
θ_0 = initial deviation angle
ϕ_0 = initial azimuthal angle
Φ' = a constant given by setting $\Phi = 0$ when $r = r_0$,

$$\Phi' = \frac{\sin\phi_0 \cos\theta_0}{(1 - \cos^2\theta_0 \cos^2\phi_0)^{\frac{1}{2}}} \sin^{-1}(1 - \cos^2\theta_0 \cos^2\phi_0)^{\frac{1}{2}} \qquad \text{(F.2)}$$

Figure F.1. is a diagram of a representative ray path. α is the deflection angle and represents the asymptotic angle of the shadow zone of the ray fan.

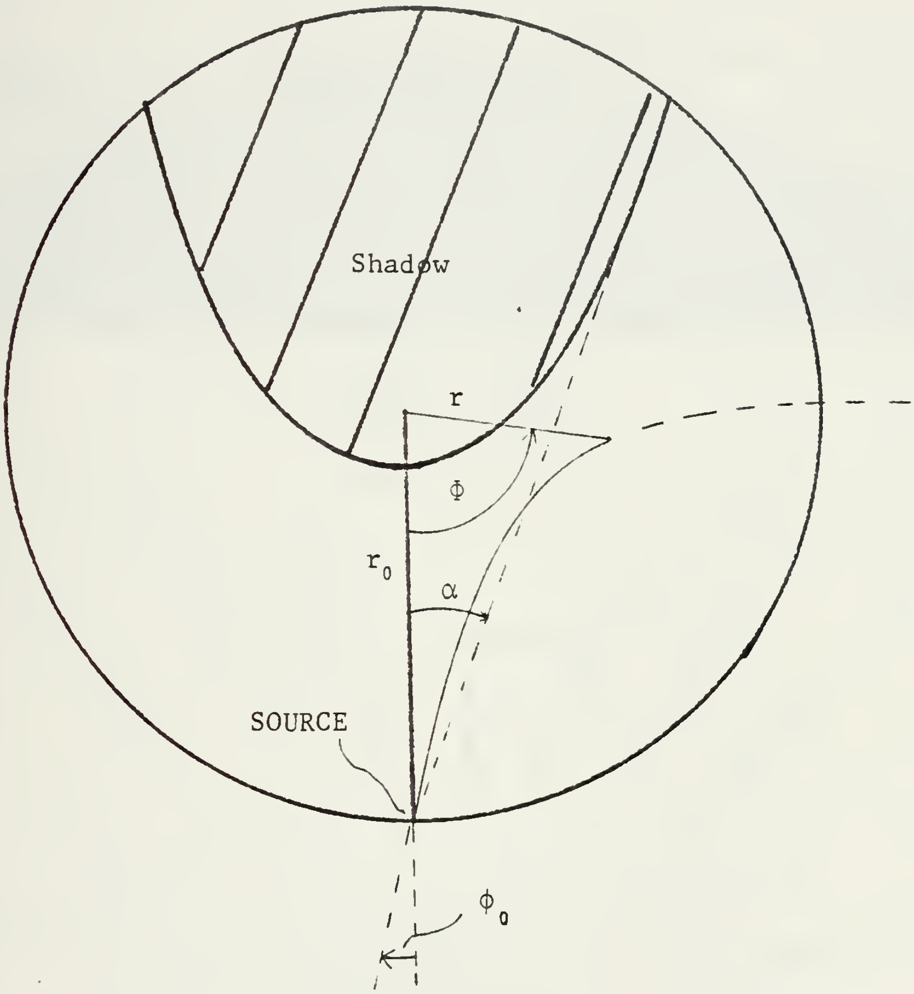


FIGURE F.1. Polar representation of the horizontal projection of a ray path for a conical seamount.

In the test run (Trial 83, Fig. F.2), the source was positioned at a range r_o of 50 km from the apex, bearing 180°, at a depth just below the surface. The receiver was placed at the apex so that for each ray the ECTRACE range r_E could be read directly from the printed ray history table values for DIST TO REC and Φ is the reciprocal of BNG TO REC. A zero gradient was used.

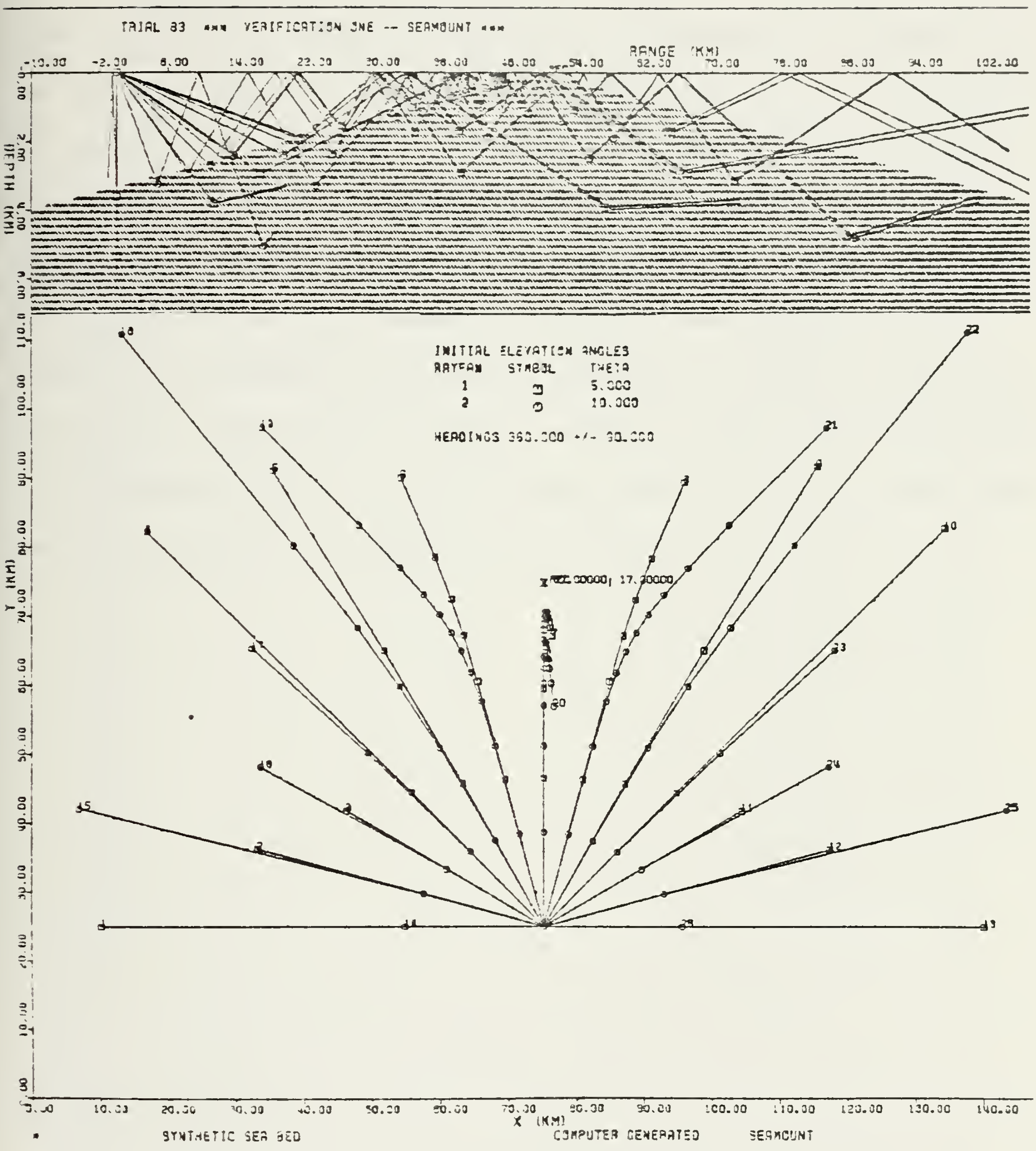


FIGURE F.2. Horizontal Plot of ECTRACE Trial for Harrison Verification.

Two ray lines were projected using θ_0= 5 degrees for ray fan 1 and θ_o = 10 degrees for ray fan 2. ϕ_0 ranged from 270° for the first ray to 90° for the last ray in each fan, with an increment of 15° between rays. Two rays were chosen for analysis, ray 6 (θ_0 = 5 degrees, ϕ_0= -15 degrees) and ray 21 (θ_0 = 10 degrees, ϕ_0 = 15 degrees). The relevant portions of their trace histories are reproduced in Table F.I and Table F.II. Only the surface reflection values are used, where the slant range DIST becomes the horizontal range r_E. Inserting θ_o and ϕ_o values into Eq. (F.1), loaded in a programmable calculator, we obtained a table of values for r for both rays. The results are shown in Table F.III and Table F.IV alongside the ECTRACE values r_E and the observed error $\varepsilon = r - r_E$. These same results are plotted in Figures F.3 and F.4 (r vs Φ).

TABLE F.I

RAY 6 (6 OF RAY FAN 1) TRACE HISTORY

REV NO.	TYPES	DEPTH (KM)	GRID HEADING	ELEV ANGLE	BRNG TO REC	DIST FM REC
0	INIT	0.001	345.000	5.000	360.000	50.000
1	BOTT	1.943	344.500	-11.825	11.370	29.200
2	SURF	0.0	344.500	11.825	22.736	21.277
3	BOTT	1.153	343.010	-16.653	34.104	17.329
4	SURF	0.0	343.010	16.653	45.508	15.170
5	BOTT	0.934	340.647	-18.460	56.912	14.035
6	SURF	0.0	340.847	18.460	68.425	13.614
7	BOTT	0.919	338.231	-17.062	79.942	13.816
8	SURF	0.0	338.231	17.062	91.462	14.689
9	BOTT	1.096	336.557	-12.677	102.980	16.464
10	SURF	0.0	336.557	12.677	114.448	19.713
11	BOTT	1.729	335.915	-6.020	125.912	25.991
12	SURF	0.0	335.915	6.020	137.458	40.960

TABLE F.II

RAY 21 (8 OF RAY FAN 2) TRACE HISTORY

REV. NO.	TYPES	DEPTH (KM)	GRID HEADING	ELEV ANGLE	BRNG TO REC	DIST FM REC
0	INIT	0.001	15.000	10.000	360.000	50.000
1	BOTT	2.451	15.649	-17.119	354.386	36.838
2	SURF	0.0	15.649	17.119	348.770	29.483
3	BOTT	1.654	17.253	-23.468	343.153	24.866
4	SURF	0.0	17.253	23.468	337.518	21.761
5	BOTT	1.303	20.043	-28.702	331.883	19.590
6	SURF	0.0	20.043	28.702	326.243	18.046
7	BOTT	1.128	24.047	-32.594	320.603	16.949
8	SURF	0.0	24.047	32.594	315.014	16.200
9	BOTT	1.046	28.867	-33.973	309.425	15.723
10	SURF	0.0	28.867	33.973	303.767	15.480
11	BOTT	1.029	34.081	-33.685	298.105	15.459
12	SURF	0.0	34.081	33.685	293.479	15.661
13	BOTT	1.071	38.566	-31.042	286.852	16.097
14	SURF	0.0	38.566	31.042	281.203	16.802
15	BOTT	1.187	41.986	-26.620	275.55	17.834
16	SURF	0.0	41.986	26.620	269.889	19.295
17	BOTT	1.420	44.169	-20.799	264.224	21.345
18	SURF	0.0	44.169	29.799	258.538	24.279
19	BOTT	1.904	45.278	-14.066	252.851	28.620
20	SURF	0.0	45.278	14.066	247.160	35.467
21	BOTT	3.160	45.674	-6.733	241.469	47.511
22	SURF	0.0	45.674	6.733	235.781	73.529

TABLE F.III

Trial 83 Ray 6

Φ	r	r_E	ε
0.0°	50.0 km	50.0 km	0.0km
-22.736	21.262	21.277	-0.015
-45.408	15.176	15.170	0.006
-68.425	13.616	13.614	0.002
-91.462	14.713	14.689	0.024
-114.49	19.822	19.713	0.109
-137.46	41.571	40.960	0.611

Ray 7 $r_{E_{ce}} = 4.339$ km ($\Phi = \phi$, constant)

$r_{ce} - r_{E_{ce}} = 0.019$ km

TABLE F.IV

Trial 83 Ray 21

Φ	r	r_E	ε
0.0°	50.0 km.	50.0 km	0.0 km
11.230	29.470	29.483	-0.013
22.482	21.745	21.761	-0.016
33.757	18.026	18.046	-0.020
44.986	16.178	16.200	-0.022
56.233	15.548	15.480	-0.022
67.521	15.643	15.661	-0.018
78.797	16.791	16.802	-0.011
90.111	19.309	19.295	0.014
101.46	24.365	24.279	0.086
112.84	35.822	35.467	0.355
124.219	75.857	73.529	2.238

Ray 20 $r_{E_{ce}} = 8.687$ km ($\Phi = 0$, constant)

$r_{ce} - r_{E_{ce}} = -0.005$ km

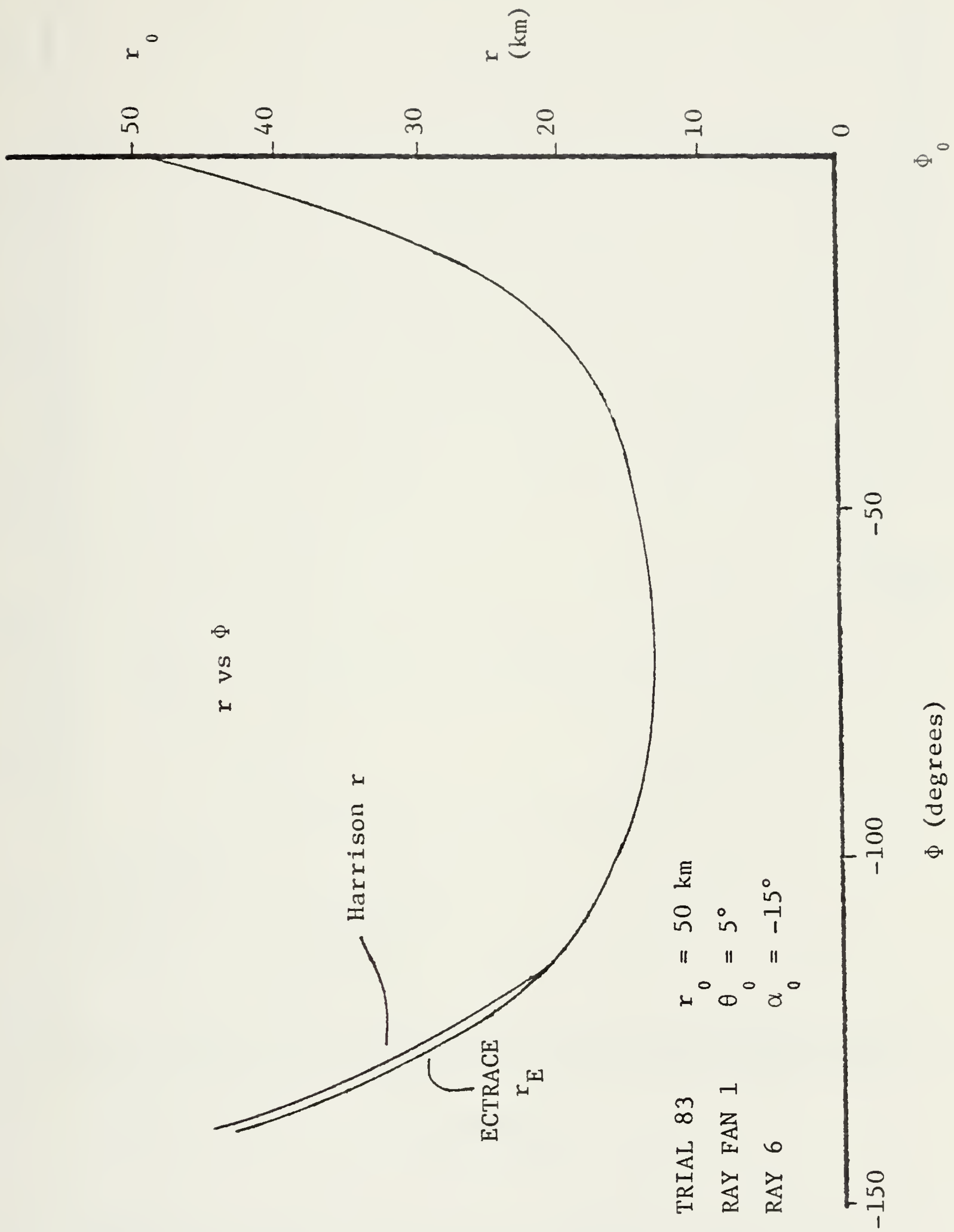


FIGURE F.3. Harrison verificaton plot of range vs Φ for ray 6.

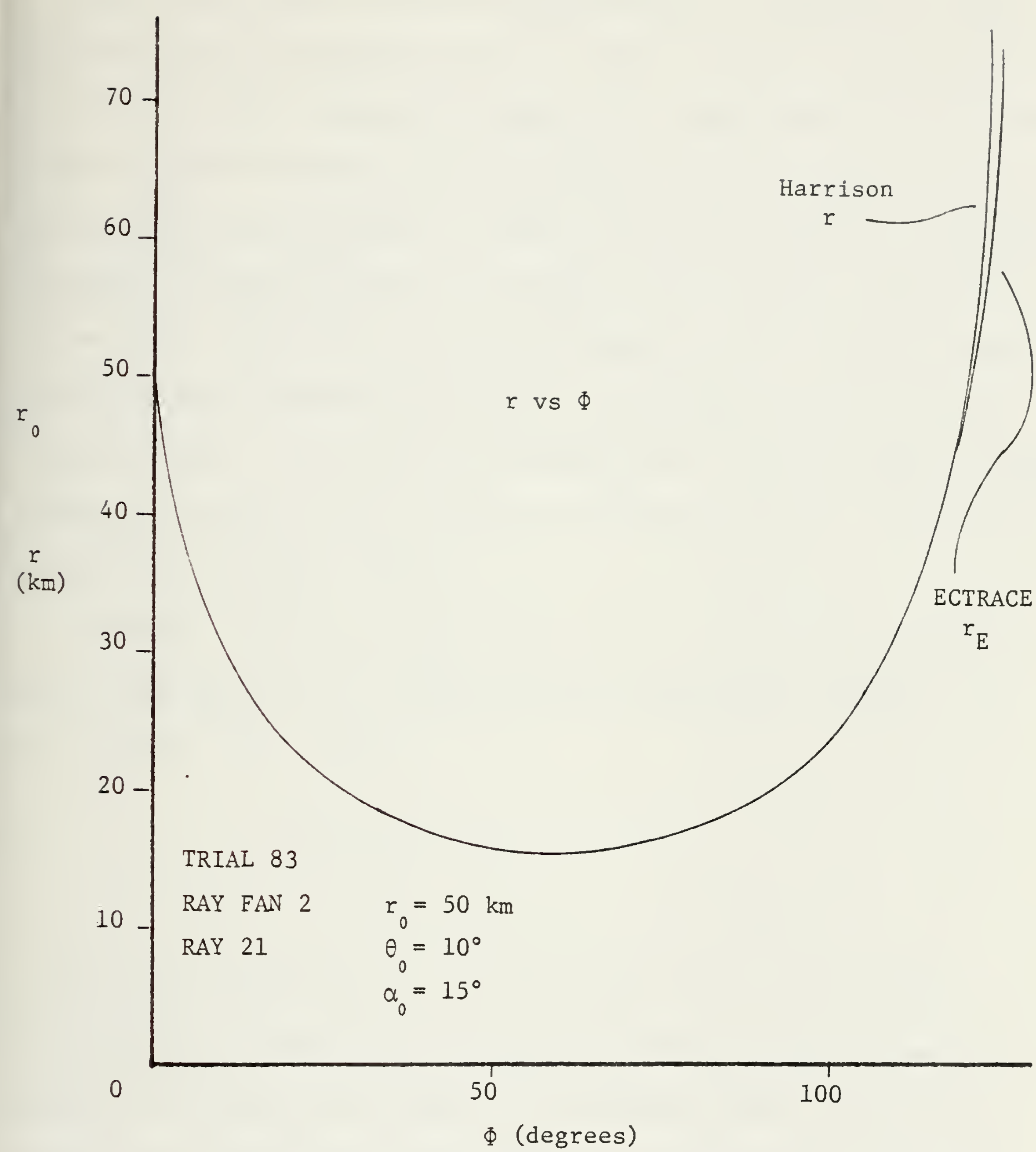


FIGURE F.4. Harrison Verification Plot of range vs Φ for ray 21

These results show very close agreement up to a distance well past the closest point of approach, and only begin to diverge slightly when the rays approach their asymptotic values for Φ. Investigation of other rays of the same trial supports this observation. Clearly the fact that the plane facets do not all have the same slope as the cone they collectively represent, but vary as much as 4% between their extremes, explains most of the error. The validity of the model decreases near the apex, where fewer data points and hence plane facets are used to approximate the conical surface. In addition, the progressively shallower grazing angles assumed by the rays after CPA cause the horizontal paths to become more sensitive to this facet effect.

Reference 13 also provides an equation for the closest approach of the ray fan envelope to the seamount center, developed by letting $\phi_0 = 0$ in Eq. (F.1). Thus

$$\begin{aligned} r_{ce} &= r_0 \sin \theta_0 \\ &= 8.682 \text{ km.} \end{aligned} \qquad (F.3)$$

Ray 21 of Trial 83 was the radial ray ($\phi_0 = 0°$) and its CPA on the printout was 8.687 km, a difference of only four meters after 11 reflections from the bottom model. The ray should have reversed its heading exactly 180 degrees after CPA, but actually experienced a 174 degree heading change, revealing the sensitivity of the model to radially-directed ray paths.

Lastly, Ref. 13 defines the asymptotic angle of the shadow zone of the ray fan (Fig. F.1). A plot of the final azimuthal angle ϕ_f versus θ_0 for ray fan 2 (Fig. F.5) of the test run exhibits an interpolated value for α of 40 degrees for $\phi_0 = 29$ degrees. Testing this value with

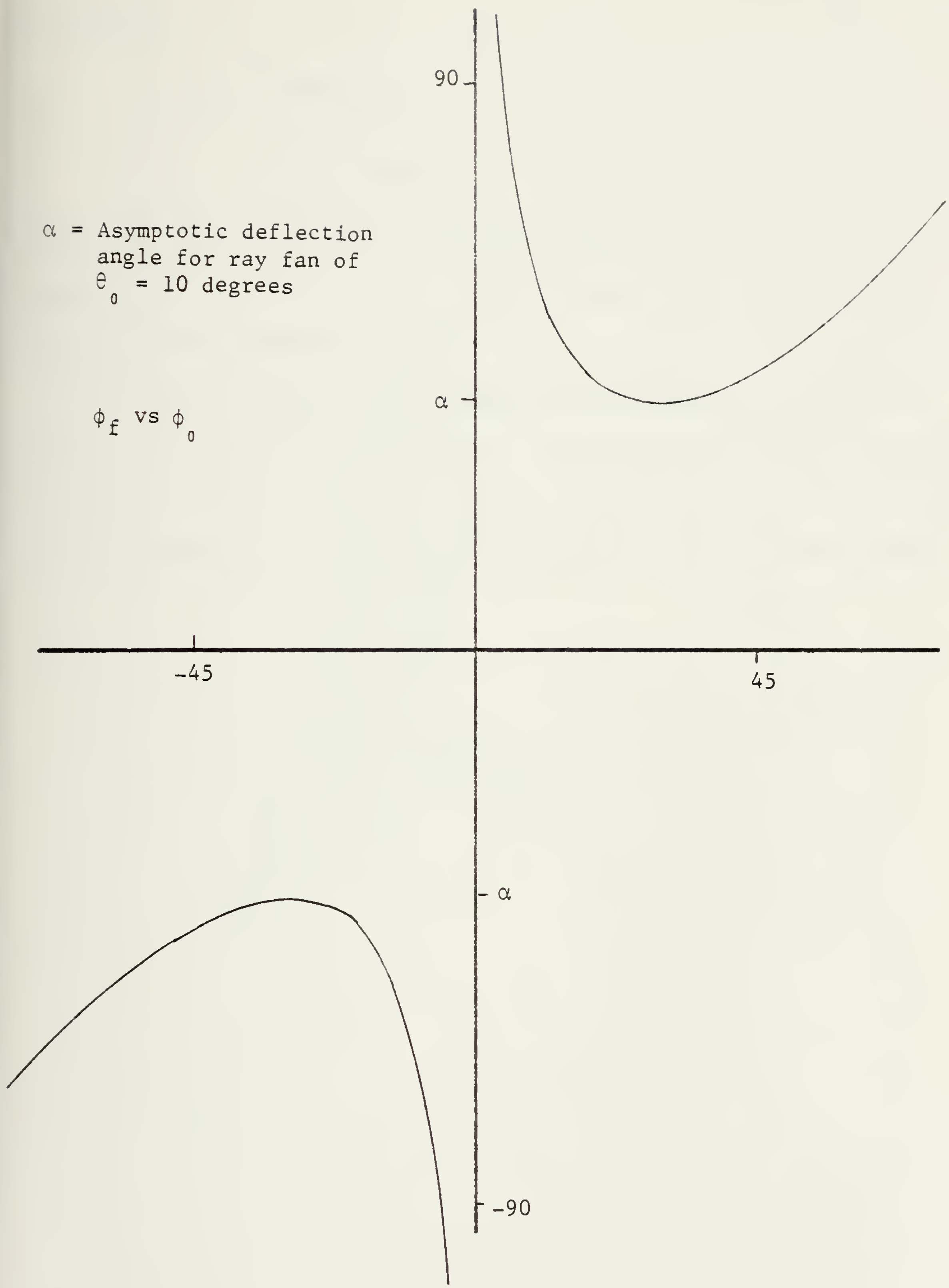


FIGURE F.5. Harrison verification plot of ϕ_f vs ϕ_0.

the formula

$$\alpha = \frac{3}{2}(\pi \sin^2\theta_0)^{1/3}$$

$$= 39.187 \text{ degrees} \qquad (F.4)$$

shows close agreement with the theoretical value. Figure F.6, an ECCOM composite for the seamount, is a graphic example of the dependence of α on the initial elevation angle.

It is worth noting that the analytical expressions for horizontal projections of ray paths apply only to exact functional bottom topographies and that the curved three dimensional ray path surfaces only approximate an actual surface of vertical planes joined at bottom reflection points.

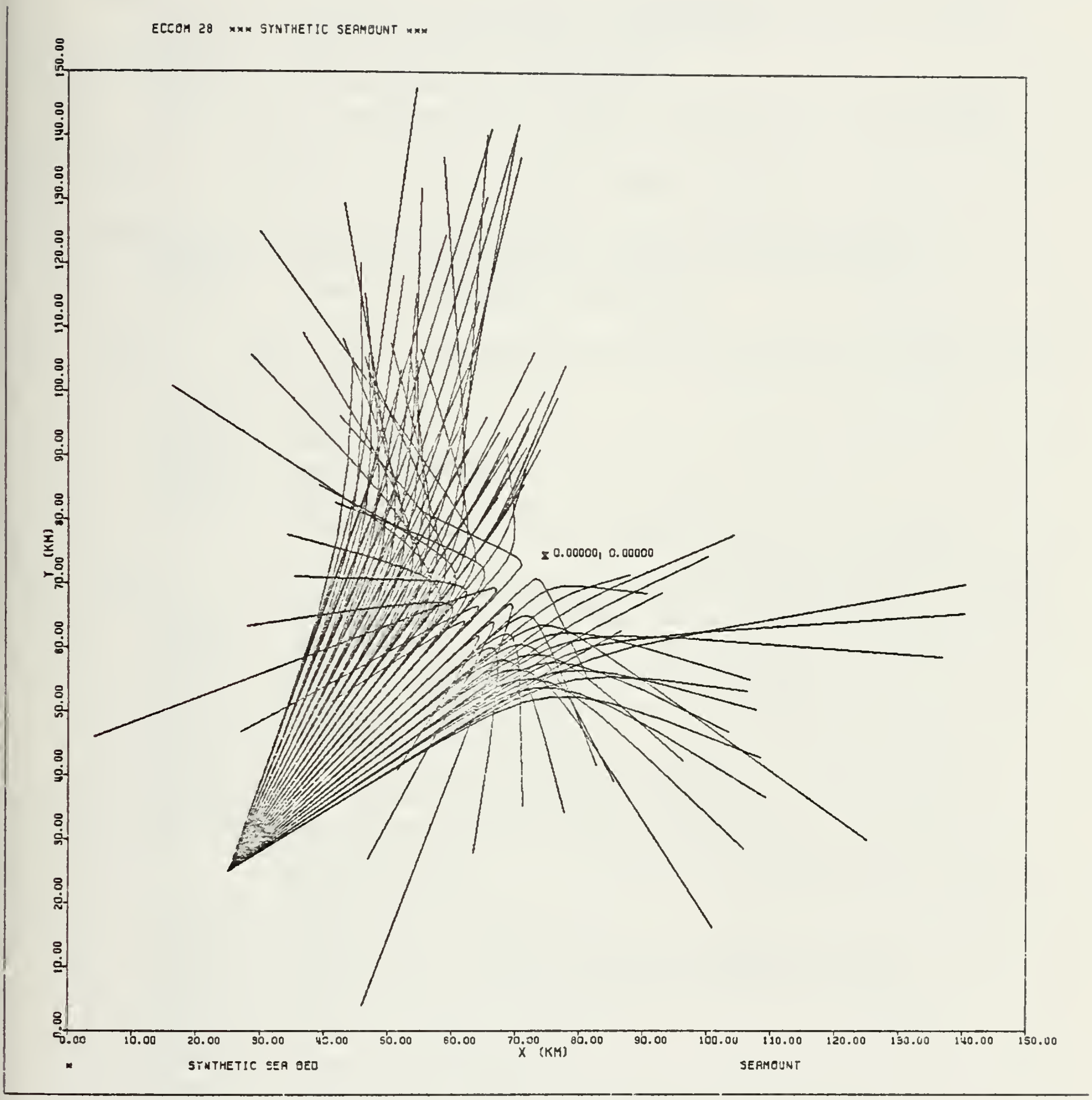


FIGURE F.6. ECCOM Trial 28 showing the effect of a synthetic conical seamount on ray propagation. The ray's initial headings varied from 20° to 58° and the initial elevation varied from 4° to 18°. Selected rays from five ECTRACE runs (100 rays) were combined.

APPENDIX G
TABLES
TABLE I
A SUMMARY OF THE PROGRAMS AND THEIR COMPONENT SUBPROGRAMS

I. GENBOT - plots source contour data and generates depth matrix
   - A. CORNER - calculates boundary coordinates
   - B. GEOPLT - plots geometric grid lines
   - C. GEODST - solves geodetic triangle
   - D. RAIN 1/2/3 - perform sorting and interpolation (from SSP3)

II. SYNGEN - produces synthetic sea bed depth matrices

III. G3DP - produces perspective surface plot of depth matrix
   - A. GRDSCT - loads working surface plot of depth matrix
   - B. PLT3D1 - draws surface plot (from NPS)

IV. GRDCHK - allows inspection of depth matrix and draws contour plot of same
   - A. GRDSCT - see above
   - B. CORNER - see above
   - C. GEOPLR - same as GEOPLT but for a rotated plot
   - D. GEODST - see above
   - E. CONTUR - draws a contour plot of depth matrix (from NPS)

V. ECTRACE - a collection of subprograms which perform ray tracing
   - A. TRACER - controls all subprograms and manages ray traces
   - B. GRDSCT - see above
   - C. IDTSUB - identifies subscripts of depth matrix
   - D. DEEP - calculates depth at given coordinates
   - E. IDPROF - identifies water mass
   - F. NUPROF - calculates water mass parameters
   - G. CHNLIM - calculates sound channel limits
   - H. CONTAC - calculates point of bottom contact
   - I. BOANG - calculates direction and grazing angle of bottom reflected ray
   - J. FBLOSS - calculates bottom loss
   - K. ANGPRT - prints ray fan bottom bounce table
   - L. BNGPLT - initializes plotting and draws plot borders
   - M. BDTPLT - draws bathymetry profile
   - N. RNGPLT - draws vertical plot
   - O. T2DPLT - draws horizontal plot
   - P. SSPPLT - draws sound speed profiles
   - Q. ENDPLT - draws plot legend and terminates plotting

   (L through Q call subroutines from the Versatec software package.)

TABLE II
INPUT PARAMETERS TABLEAU (COMPUTER PRINT OUTPUT)

ECTRACE TRIAL 67 *** SYNTHETIC BASIN ***

SOUND SPEED PROFILES

ZC	C(ZC)	GRAD
0.0	1461.08	0.0
50.00	1461.00	-1.40000
60.00	1447.00	-0.20000
70.00	1445.00	0.10000
90.00	1447.00	0.01780
500.00	1454.30	0.01218
1050.00	1461.00	0.02000
1500.00	1470.00	0.01668
4000.00	1511.70	0.01668

NWMR (Northwest Mohns Ridge)

ZC	C(ZC)	GRAD
0.0	1490.00	-0.16000
50.00	1482.00	-0.05000
200.00	1474.50	-0.00286
375.00	1474.00	-0.02133
750.00	1466.00	-0.00444
975.00	1465.00	0.00727
1250.00	1467.00	0.01200
1500.00	1470.00	0.01668
4000.00	1511.70	0.01668

SEMR (Southeast Mohns Ridge)

ZC	C(ZC)	GRAD
0.0	1470.00	0.0
1500.00	1470.00	0.01668
4000.00	1511.70	0.01668

ISOV (Isovelocity)

(Table II continued)

continued TABLE II

```
FRONT/BOUNDARY INTERSECTIONS
          FRONT                                            FRONT
  ENTRY            EXIT                  ENTRY                  EXIT
( 0.0, 50.0)  (100.0,150.0)           ( 50.0, 0.0)        (150.0, 100.0)

GRID FILE CALLED  *****************************************************
                  *                                                   *
                  *       SYNTHETIC SEA BED                           *
                  *       COMPUTER GENERATED  BASIN                   *
                  *                                                   *
                  *****************************************************
HAD BEEN READ.  CENTER LAT/LONG IS  70.07059/ 15.85032 AT
(75.000, 75.000) KM. RANGES ARE  0.0 TO 150.0 KM IN Y (22801 ITEMS).

THE FOLLOWING ADDITIONAL PARAMETERS WERE INPUT:

SOURCE DATA:  FREQUENCY 150.00 HZ,GRID COORDINATES ( 40.00,40.00),
DEPTH  0.850 KM.  8 RAY FANS WILL BE TRACED, FROM -30.0 TO 30.0 DEGREES
ELEVATION, INCREMENTING EVERY  8.6 DEG.  1 RAYS MAKE A RAY FAN FROM  10.0
TO  10.0 DEGREES TRUE HEADING, INCREMENTING EVERY  0.0 DEGREES.

RECEIVER DATA:  GRID COORDINATES ( 125.0,  40.0), DEPTH 1.749 KM.

GRID ARRAY:  MAXIMUM ARRAY SIZE (151,151).DEPTH VALUES RESOLVED BY  1.0
             KM. SPACING.

TRACE PARAMETERS:  PROPAGATION WILL CONTINUE UNTIL A BOTTOM BOUNCE LOSS
                   OF 150.0 DB OR  21 BOTTOM CONTACTS EXCEEDED FOR EACH
                   RAY. PATH INCREMENT IS  100.0 M. POSITION ERROR AFTER
                   BOTTOM CONTACT WITHIN  0.05000 KM.  CHANNEL RANGE
                   LIMIT 200.0 KM. SMALL ANGLE 2.0 DEG.

PLOT OPTION SELECTED.
PLOT PARAMETERS:           X          Y          Z          R
        AXES:             0.0        0.0       -1.0       -10.0
                         10.0       10.0        2.0         7.00
                        150.0      150.0        6.0       123.00
     SCALING:  LENGTH 4 IN., FACTOR=0.2667

EACH SEGMENT REPRESENTS 10 POSITION CALCULATIONS.
```

(program parameters are user supplied via DATA cards)

TABLE III

Ray History Tableau ECTRACE Trail 67 (computer print output)
Ray 8 (of Ray Fan 8) TRACE HISTORY

REV. NO.	TYPE	X(KM)	Y(KM)	DEPTH(KM)	HEADING	ELEV. ANG.
0	INIT	40.000	40.000	0.850	10.000	30.000
1	BOTT	40.722	44.097	3.161	12.533	-20.323
2	SURF	42.397	51.631	0.0	12.533	21.104
3	BOTT	44.611	61.591	4.043	14.157	-14.658
4	SURF	47.628	73.549	0.0	14.157	17.615
5	BOTT	50.933	86.652	4.344	15.556	-14.938
6	SURF	54.384	99.050	0.0	15.556	18.425
7	BOTT	57.154	109.001	3.628	17.472	-22.313
8	SURF	59.508	116.479	0.0	17.472	26.133
9	BOTT	61.260	122.046	2.780	20.496	-31.868
10	SURF	62.747	126.024	0.0	20.496	33.698
11	BOTT	63.904	129.119	2.190	25.415	-41.798
12	SURF	64.930	131.279	0.0	25.415	42.680
13	BOTT	65.778	133.064	1.823	33.460	-51.533
14	SURF	66.569	134.260	0.0	33.460	51.978
15	BOTT	67.260	135.305	1.606	46.953	-60.129
16	SURF	67.933	135.934	0.0	46.953	60.372
17	BOTT	68.551	136.511	1.490	69.313	-65.773
18	SURF	69.182	136.749	0.0	69.313	65.930
19	BOTT	69.787	136.977	1.452	97.778	-66.103
20	SURF	70.428	136.890	0.0	97.778	66.244
21	BOTT	71.073	136.802	1.482	121.284	-61.160
22	SURF	71.771	136.378	0.0	121.284	61.347
23	BOTT	72.510	135.929	1.587	135.723	-53.031
24	SURF	73.338	135.080	0.0	135.723	53.338
25	BOTT	74.265	134.129	1.786	144.372	-43.592
26	SURF	75.339	132.631	0.0	144.372	44.152
27	BOTT	76.618	130.845	2.127	149.526	-34.065
28	SURF	78.158	128.228	0.0	149.526	35.164
29	BOTT	80.046	125.020	2.664	153.024	-26.290
30	SURF	82.376	120.443	0.0	153.024	26.244
31	BOTT	85.445	114.414	3.452	155.238	-19.001
32	SURF	89.145	106.392	0.0	155.238	20.346
33	BOTT	93.855	96.179	4.247	156.763	-15.037
34	SURF	98.821	84.615	0.0	156.763	18.321
35	BOTT	103.807	73.002	4.261	158.297	-17.529
36	SURF	107.968	62.549	0.0	158.297	20.418
37	BOTT	111.257	54.284	3.486	160.557	-25.605
38	SURF	113.497	47.940	0.0	160.557	28.126
39	BOTT	115.211	43.084	2.713	164.103	-35.243
40	SURF	116.231	39.501	0.0	164.103	36.331
41	BOTT	117.049	36.630	2.188	169.812	-45.355

BOTTOM LOSS GATE EXCEED. TERMINATING RAYTRACE AFTER 3311 SEGMENTS. PLOT BASED ON 367 POINTS, 41 REVERSALS AND 21 BOTTOM BOUNCES. MINIMUM DISTANCE TO RECEIVER, 8.57 KM, OCCURS AT TIME 166.244

(TABLE III continued)

continued TABLE III

BRING TO SRC*	DIST FM SRC*	TIME	WTR MASS	LYR	DEPTH MIN	LIMITS MAX
0.0	0.0	0.0	SEMR	5	0.0	*****
190.00	4.759	3.270	SEMR	8	0.0	*****
191.646	11.906	8.802	SEMR	1	0.0	*****
192.056	23.308	16.265	SEMR	9	0.0	7.093
192.809	34.416	24.931	SEMR	1	0.0	7.093
193.189	48.043	34.520	SEMR	9	0.0	7.526
193.690	60.783	43.604	SEMR	1	0.0	7.526
193.961	71.155	51.036	NWMR	8	0.0	*****
194.310	78.932	56.802	NWMR	1	0.0	*****
194.572	84.778	61.279	NWMR	8	0.0	*****
194.812	88.984	64.648	NWMR	1	0.0	*****
195.015	92.279	67.440	NWMR	8	0.0	*****
195.276	94.626	69.600	NWMR	1	0.0	*****
195.483	96.573	71.568	NWMR	8	0.0	*****
195.741	97.936	73.103	NWMR	1	0.0	*****
195.962	99.130	74.624	NWMR	8	0.0	*****
196.234	99.921	75.814	NWMR	1	0.0	*****
196.480	100.647	77.058	NWMR	7	0.0	*****
196.784	101.058	78.109	NWMR	1	0.0	*****
197.074	101.451	79.258	NWMR	7	0.0	*****
197.435	101.559	80.307	NWMR	1	0.0	*****
197.796	101.669	81.494	NWMR	7	0.0	*****
198.245	101.483	82.550	NWMR	1	0.0	*****
198.721	101.290	83.914	NWMR	8	0.0	*****
199.322	100.758	85.183	NWMR	1	0.0	*****
200.003	100.176	86.827	NWMR	8	0.0	*****
200.882	99.146	88.467	NWMR	1	0.0	*****
201.954	97.956	90.673	NWMR	8	0.0	*****
203.388	96.130	93.144	NWMR	1	0.0	*****
205.222	93.996	96.346	SEMR	8	0.0	*****
207.779	90.926	100.139	SEMR	1	0.0	*****
211.412	87.232	105.373	SEMR	8	0.0	*****
216.510	82.606	111.713	SEMR	1	0.0	*****
223.790	77.898	110.854	SEMR	9	0.0	7.469
232.820	73.832	128.722	SEMR	1	0.0	7.469
242.651	71.917	137.727	SEMR	9	0.0	8.687
251.646	71.615	145.718	SEMR	1	0.0	8.687
258.664	72.722	152.236	ISOV	2	0.0	*****
263.834	73.929	157.358	ISOV	1	0.0	*****
267.652	75.297	161.318	ISOV	2	0.0	*****
270.375	76.237	164.413	ISOV	1	0.0	*****
272.504	77.134	166.955	ISOV	2	0.0	*****

*For this run range and bearing of the rays were referenced to the source. As explained in Chapter V, in the current version of ECTRACE, these values are now referenced from the receiver (REC).

(TABLE III continued)

continued TABLE III

C (KM/S)	GRAZ ANGLE	BTM ANGLE	BTM LOSS	LPS
1.46556				
1.49770	24.081	4.678	4.1	0
1.49000				
1.51242	16.731	3.392	6.1	0
1.49000				
1.51743	14.465	2.756	7.2	0
1.49000				
1.50549	19.479	3.935	10.0	1
1.46100				
1.49135	27.785	5.015	15.1	1
1.46100				
1.48151	37.224	5.651	22.2	1
1.46100				
1.47539	46.949	6.015	30.0	0
1.46100				
1.47177	56.159	6.182	40.9	0
1.46100				
1.46980	63.522	6.272	50.9	0
1.46100				
1.46904	66.735	6.262	60.0	0
1.46100				
1.46963	64.227	6.248	70.9	0
1.46100				
1.47145	57.381	6.141	80.9	0
1.46100				
1.47477	48.418	6.040	89.9	0
1.46100				
1.48045	38.786	5.639	97.3	0
1.46100				
1.48941	30.571	5.162	103.0	2
1.49000				
1.50255	22.176	4.155	106.6	0
1.49000				
1.51582	16.287	2.886	108.3	0
1.49000				
1.51606	16.298	2.906	110.1	1
1.49000				
1.50313	22.375	4.256	113.8	0
1.47000				
1.49023	30.985	5.212	119.5	0
1.47000				
1.48147	40.627	5.821	127.3	0

TABLE IV

Ray fan Summary Tableau for ECTRACE Trail 67 ray number 8. (computer print output)

ANGLES BEFORE BOUNCE. INITIAL ELEVATION ANGLE = 30.00

HEADINGS:	10.00
BOUNCE	
1	27.78
2	18.77
3	13.96
4	16.60
5	23.63
6	32.51
7	42.10
8	51.66
9	60.18
10	65.80
11	66.12
12	61.15
13	52.96
14	43.36
15	34.76
16	25.30
17	17.50
18	15.03
19	19.09
20	26.64
21	35.75

DEPTH IN METERS AT BOUNCE. INITIAL ELEVATION ANGLE = 30.00

HEADINGS:	10.00
BOUNCE	
1	3160.6
2	4042.9
3	4343.8
4	3627.9
5	2780.2
6	2190.1
7	1823.4
8	1606.3
9	1490.1
10	1452.0
11	1481.7
12	1586.9
13	1785.7
14	2126.5
15	2663.6
16	3451.5
17	4246.8
18	4261.3
19	3486.1
20	2713.1
21	2187.8

TABLE V

CPA Summary Tableau for ECTRACE Trial 67 (computer printed output)

8 RAY TRACES COMPLETED

H HEADING AT CLOSEST APPROACH TO RECEIVER
T TIME AT CLOSEST APPROACH TO RECEIVER
D MINIMUM DISTANCE TO RECEIVER

RAY NUMBER	INITIAL HEADING	INITIAL ELEVATION	NUMBER OF BOUNCES	H	T	D
1	10.000	-30.000	21	164.266	166.534	7.888
2	10.000	-21.429	21	158.372	165.018	1.339
3	10.000	-12.857	21	146.487	163.575	6.884
4	10.000	-4.286	21	145.103	160.945	7.175
5	10.000	4.286	12	144.811	160.593	7.803
6	10.000	12.857	21	149.200	163.395	6.676
7	10.000	21.429	21	157.810	163.22	1.139
8	10.000	30.000	21	164.103	166.244	8.575

TABLE VI

RAY FAN SYMBOLS - used to denote location of bottom reflections in the ECTRACE plots. Thirteen different symbols are used before repetition. All rays of the same ray fan (initial elevation angle) will indicate a bottom reflection with the same symbol. Each ECTRACE plot lists the symbols used for the horizontal and vertical plots.

INITIAL ELEVATION ANGLES

RAYFAN	SYMBOL	THETA
1	□	-10.000
2	○	-9.000
3	△	-8.000
4	+	-7.000
5	×	-6.000
6	◇	-5.000
7	↑	-4.000
8	⊼	-3.000
9	Z	-2.000
10	Y	-1.000
11	⌧	0.000
12	✳	1.000
13	⧖	2.000
14	□	3.000
15	○	4.000
16	△	5.000
17	+	6.000
18	×	7.000
19	◇	8.000
20	↑	9.000
21	⊼	10.000

HEADINGS 90.000 +/- 0.000

TABLE VII

ECCOM printed output. The rays which are plotted are listed, and their initial heading, initial elevation, and the number of points (initial plus reflections) are indicated. In this run the plot produced is shown in Fig. 7.

ECCOM 51 *** SYNTHETIC BASIN ***
SYNTHETIC SEA BED
COMPUTER GENERATED

XST	XDV	YST	YDV	CNX	CNY	BASIN CNLAT	CNLON
0.0	10.000	0.0	10.000	75.000	75.000	0.0	0.0

SCALE FACTOR= 0.340

INTIAL ELEVATION	INTIAL HEADING	POINTS
40.000	40.000	21
40.000	44.000	21
40.000	48.000	21
38.000	42.000	21
38.000	46.000	21
36.000	40.000	21
36.000	44.000	21
36.000	48.000	21
34.000	42.000	21
34.000	46.000	21
60.000	40.000	21
60.000	44.000	21
60.000	48.000	21
51.333	42.000	21
51.333	46.000	21
42.667	40.000	21
42.667	44.000	21
42.667	48.000	21
34.000	42.000	21
34.000	46.000	21
60.000	5.000	21
60.000	9.000	21
60.000	13.000	21
62.000	7.000	21
62.000	11.000	21
64.000	5.000	21
64.000	9.000	21
64.000	13.000	21
66.000	7.000	21
66.000	11.000	21

APPENDIX H
FIGURES

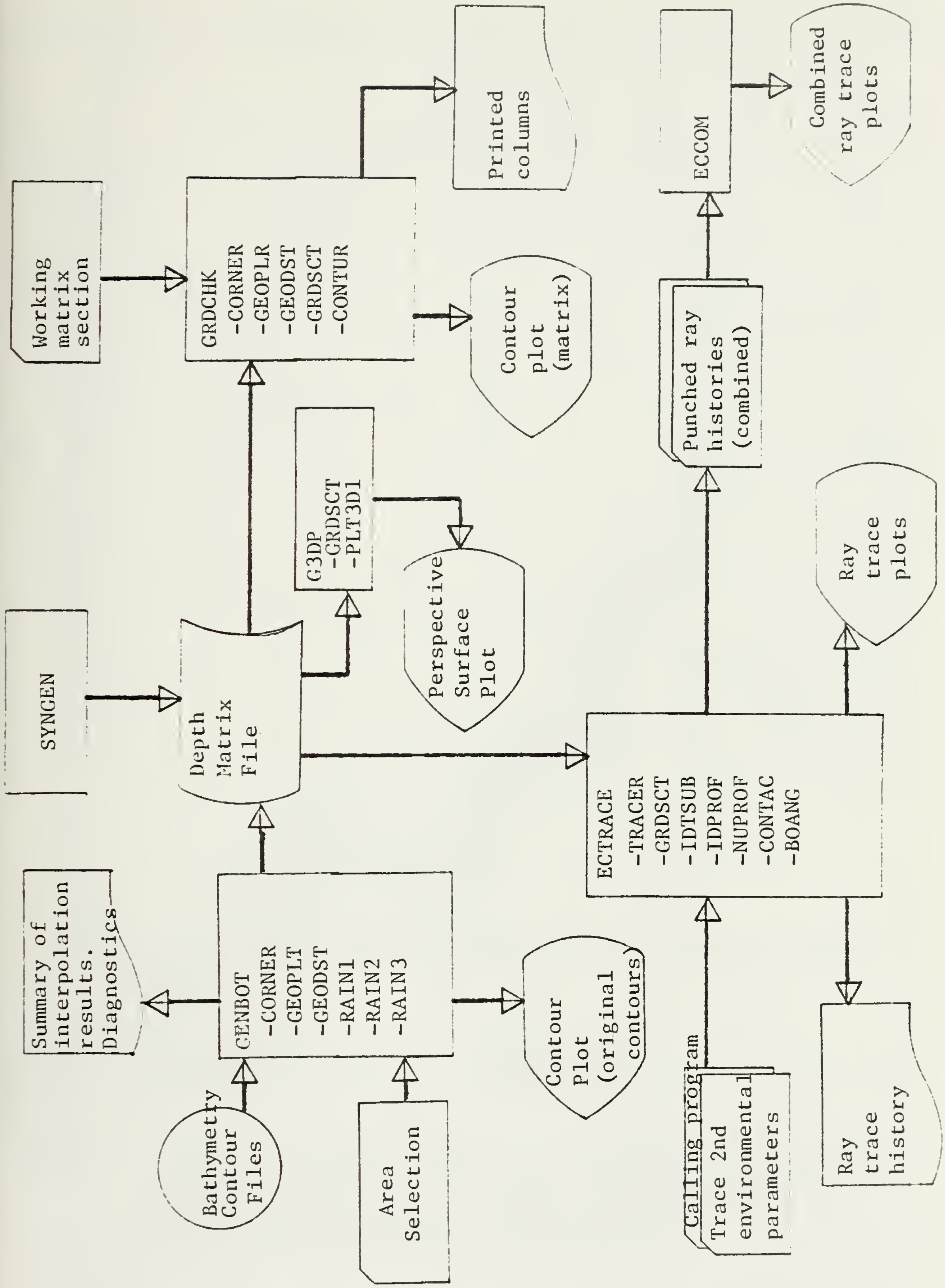


FIGURE 1. Program and Data Utilization Flowchart.

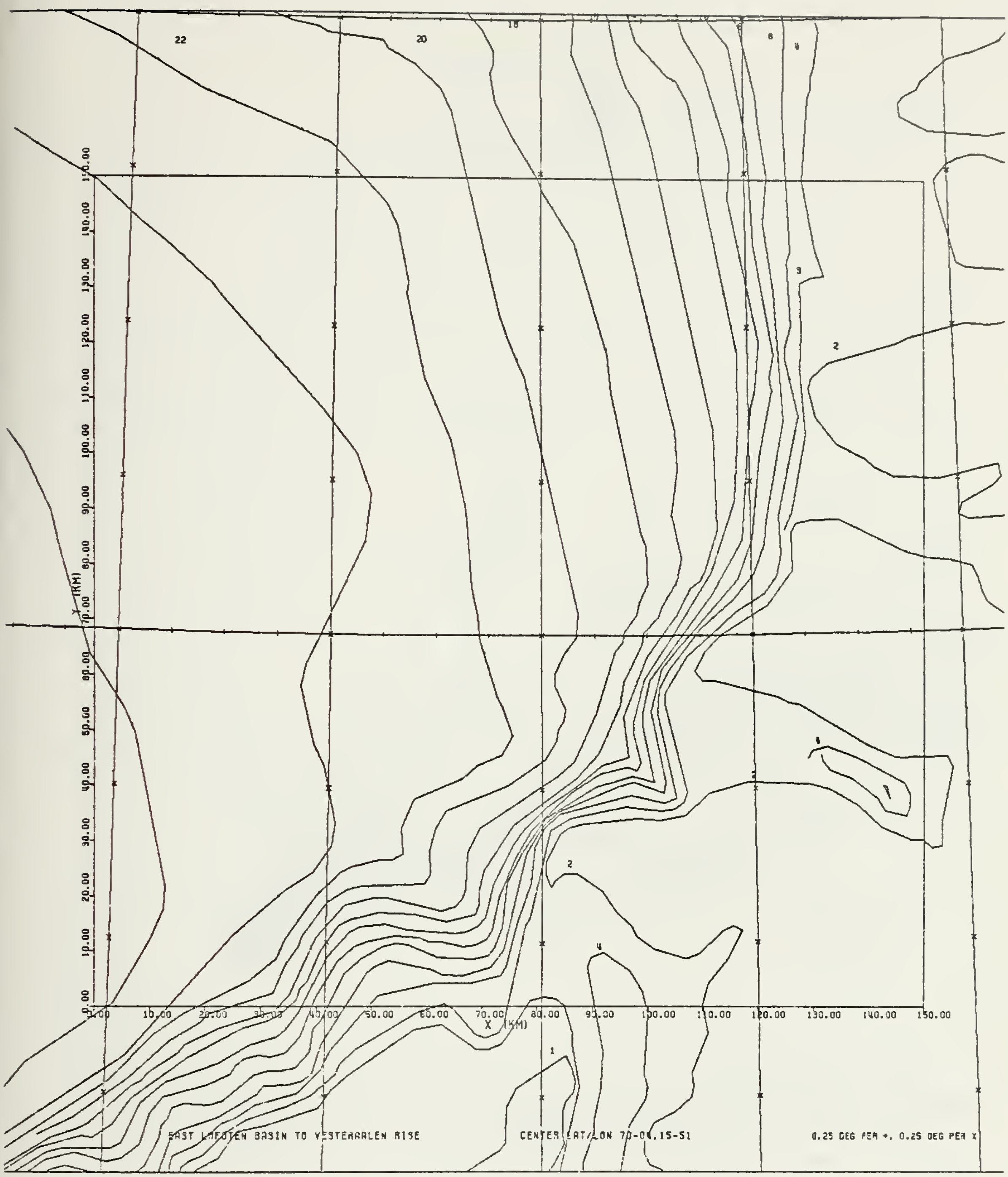


FIGURE 2. A contour plot of the eastern Lofoten Basin produced by GENBOT from magnetic tape data. The rectantular labeled axis denotes the region for which GENBOT generated a matrix representing evenly spaced depth values. A geographic coordinate grid is superimposed.

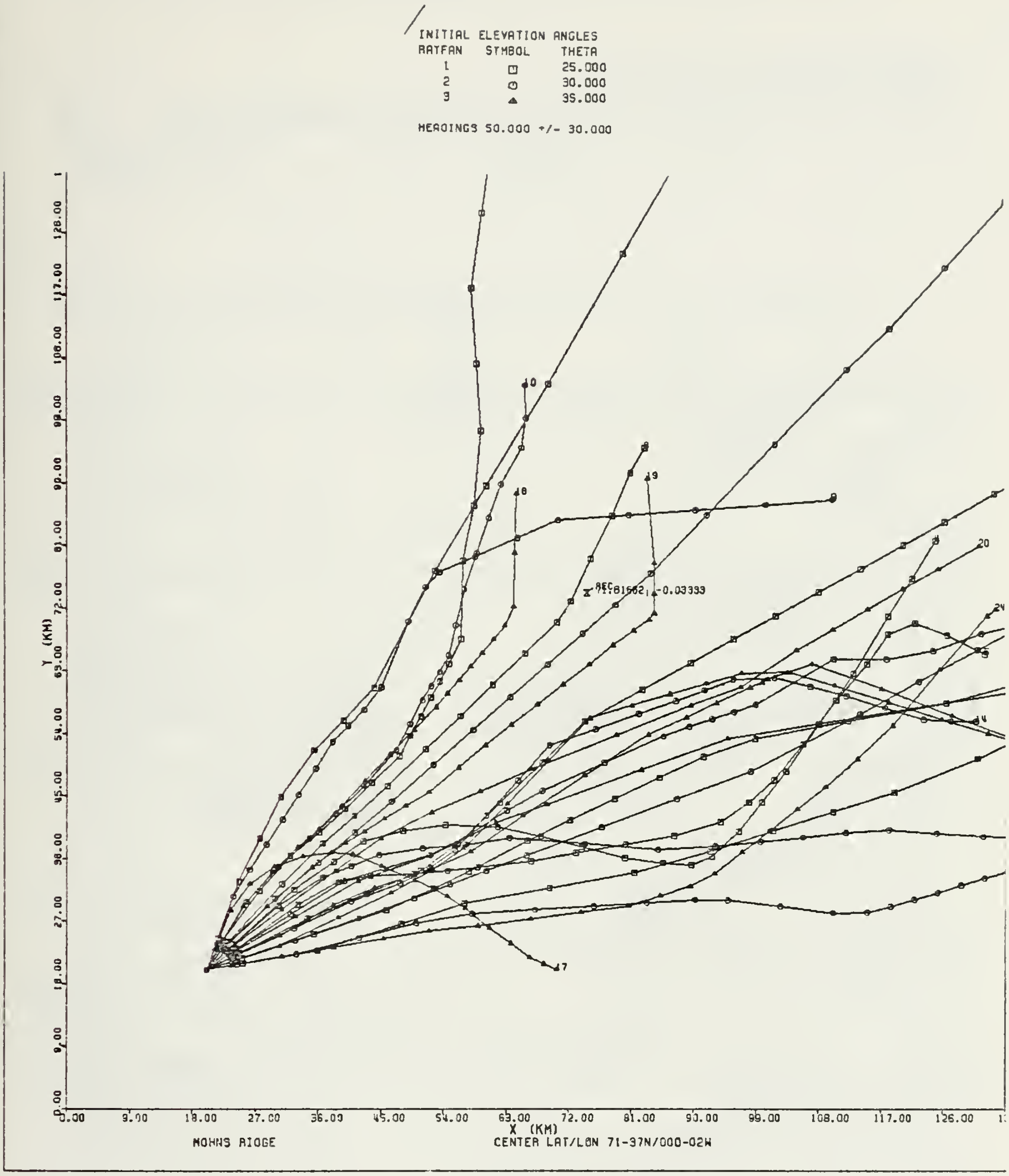


FIGURE 3. ECTRACE TRIAL 82C performed over a depth-grid matrix of the Mohns Ridge. In this run three ray fans are used with eight rays per ray fan. This may be overlaid on a contour plot of this area (Fig. 2).

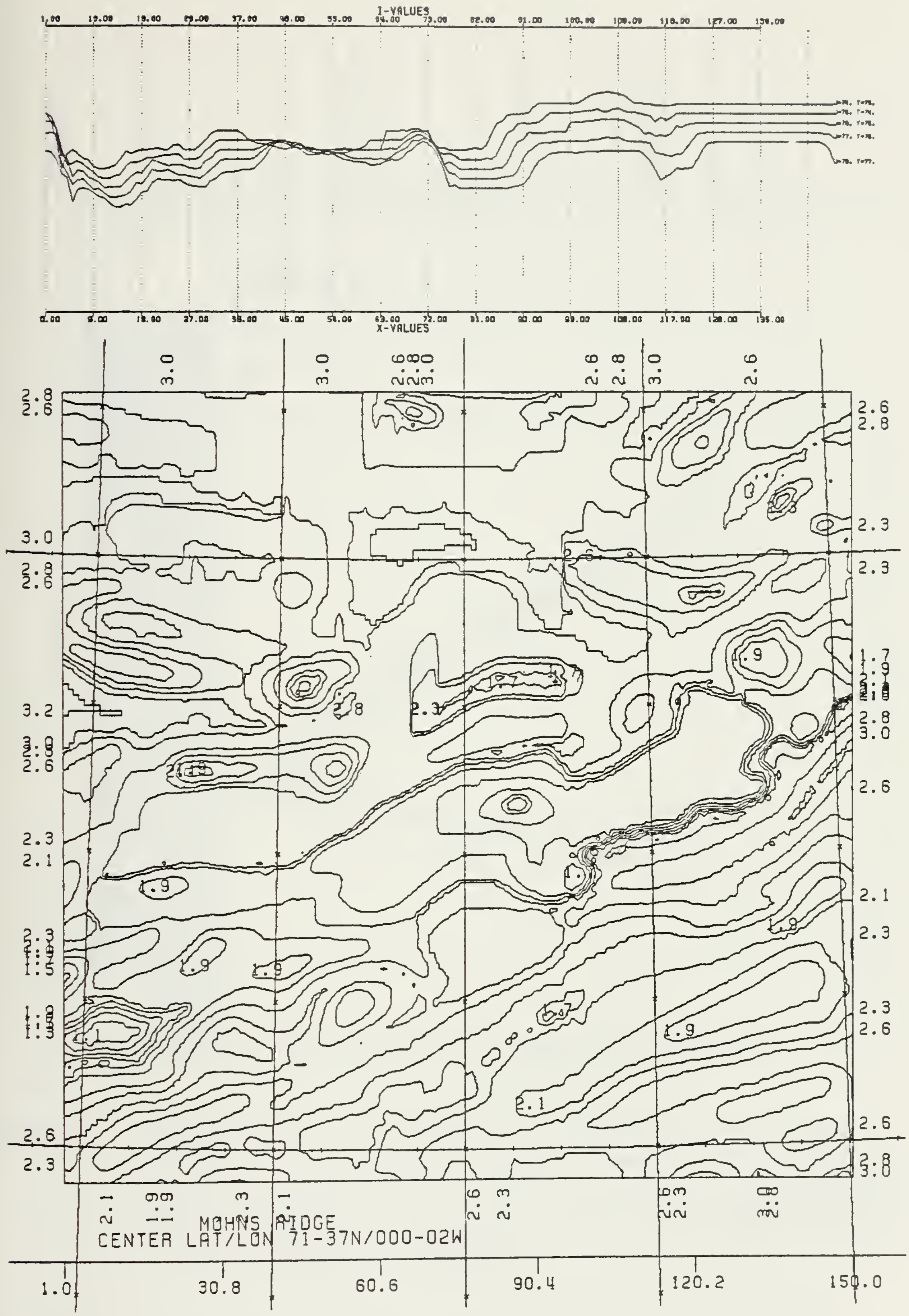


FIGURE 4. A GRDCHK plot made on a matrix generated from contour data. In this example, rows 1 through 150 of columns 74 through 78 were selected for profile analysis and printing.

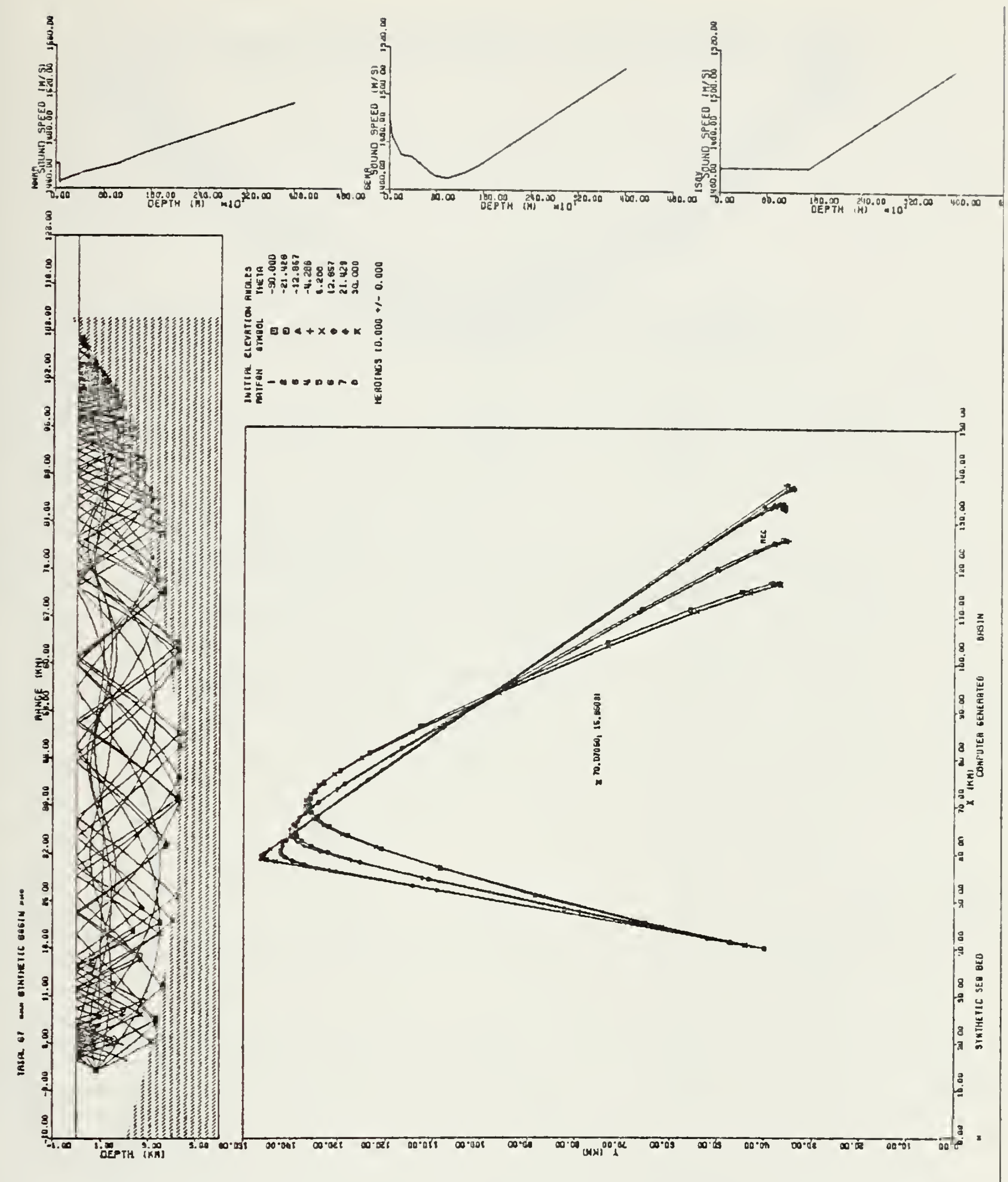


FIGURE 5. ECTRACE TRIAL 67, performed in a synthetic parabolic basin (FIGURE 6). In this run eight ray fans are used with initial elevation angles varying between -30 and +30 degrees. Since there is only one ray in each ray fan, the initial heading for the rays will be 10 degrees. The vertical plot reveals CZ, sofar ducted, and SRBR rays.

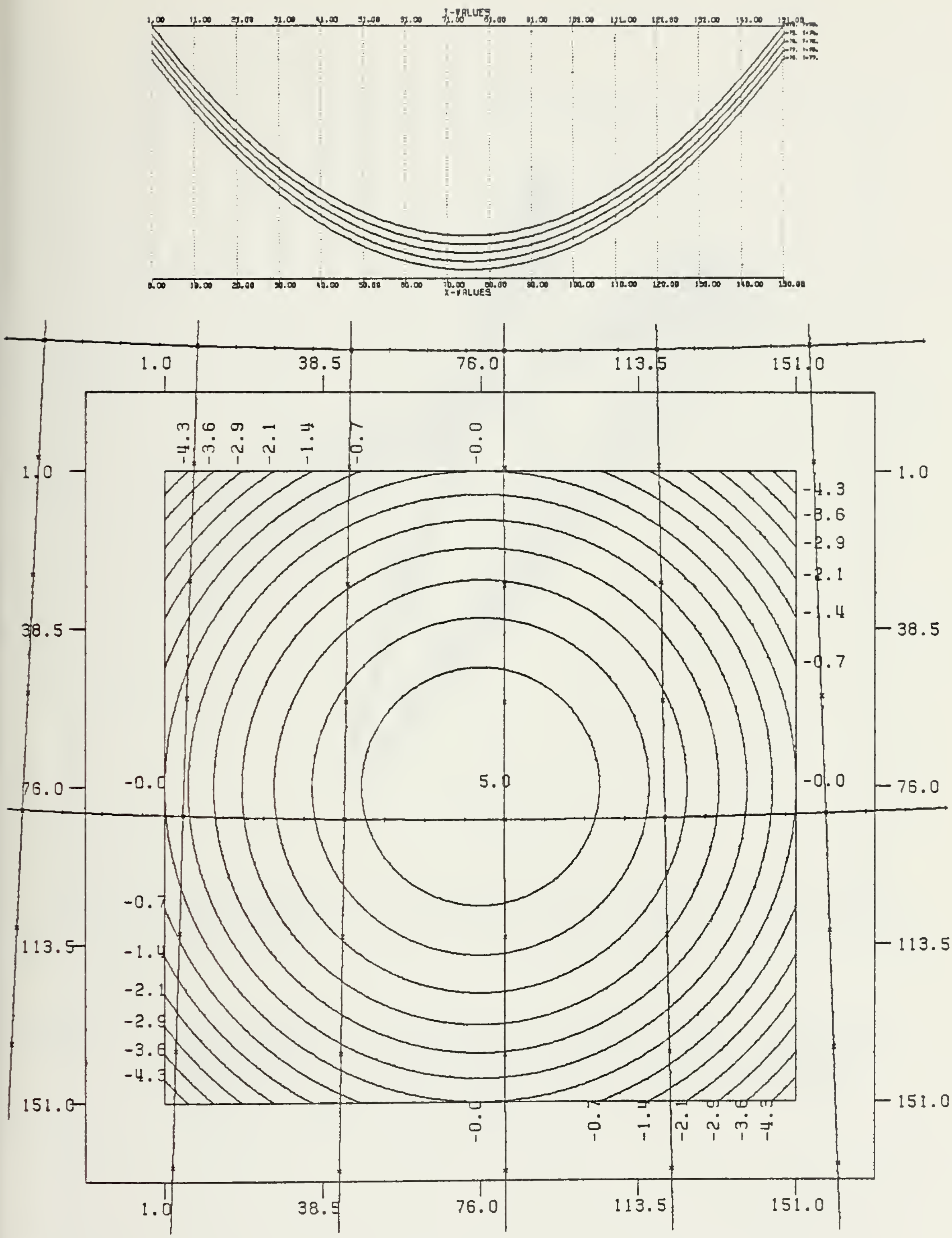


FIGURE 6. A graphical analysis performed by GRDCHK of a synthetic parabolic basin (Fig. 10) created by SYNGEN. The upper plot is a series of displaced depth curves of selected columns of the matrix.

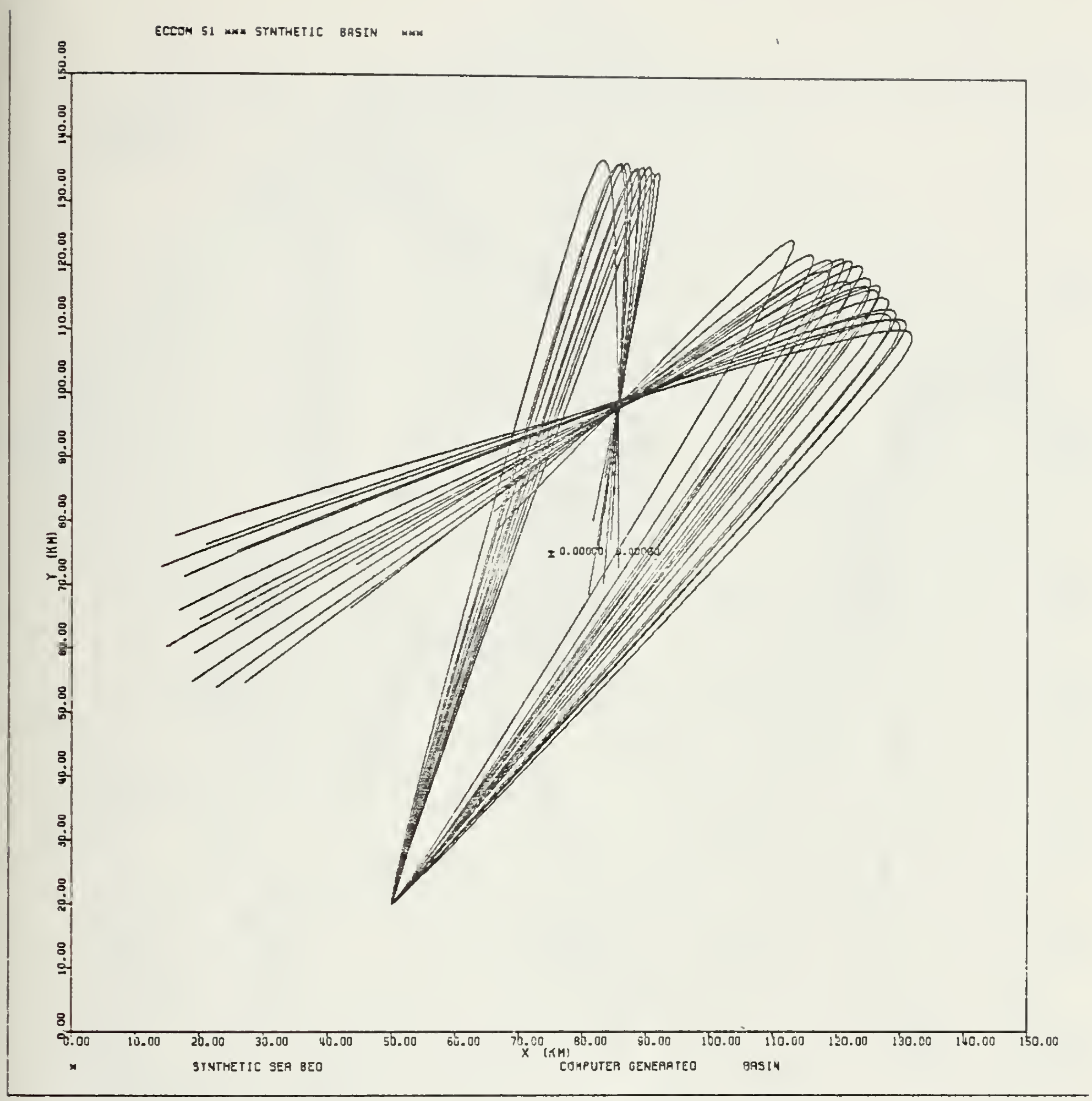


FIGURE 7. Convergence in a parabolic basin (Fig. 10). This is an ECCOM product using punched output from three separate ECTRACE runs. TABLE VII identifies the ray headings, elevation angles, and number of points.

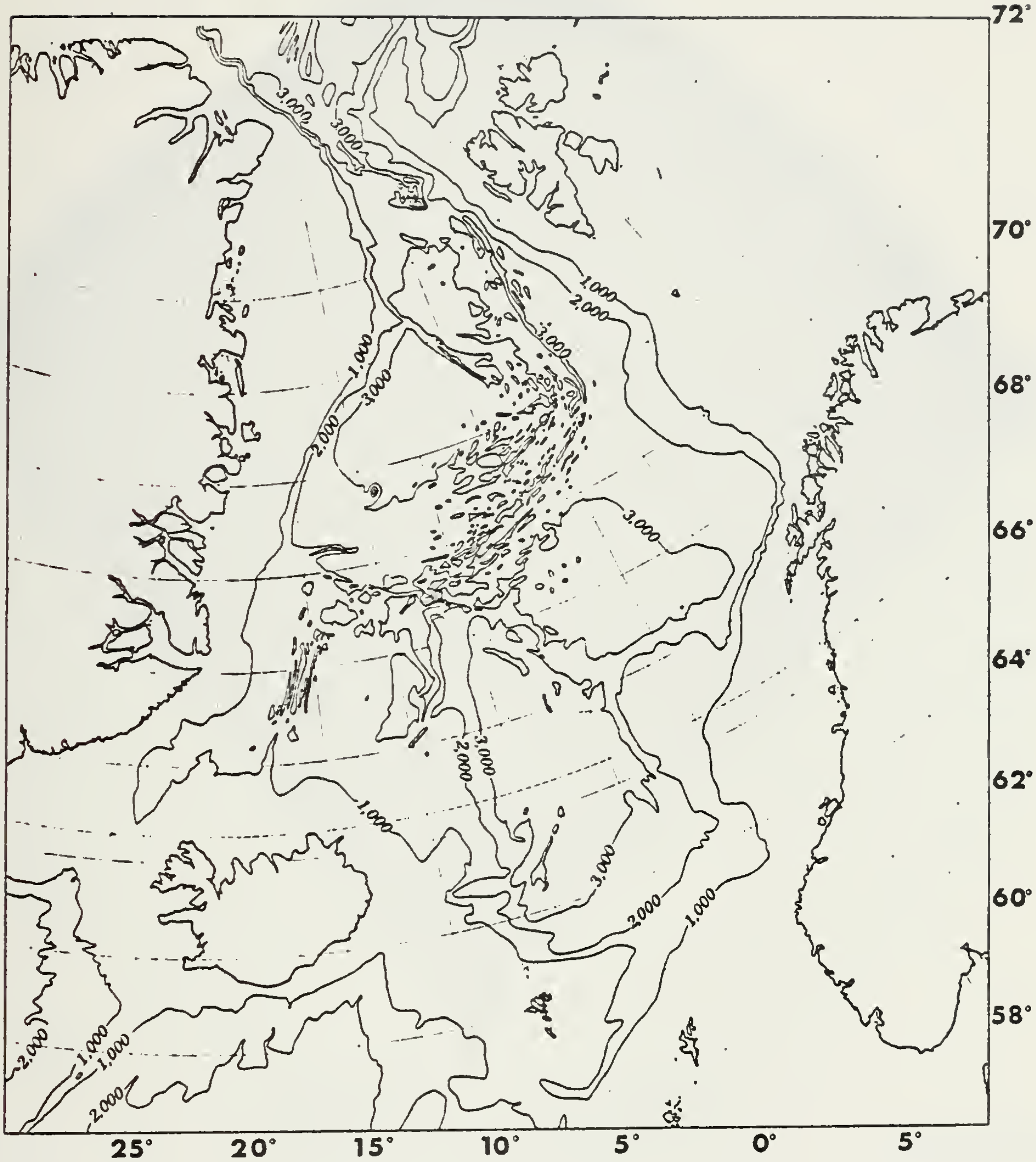


FIGURE 8. A reduced representation of a chart titled "Bathymetry of the Norwegian-Greenland and Western Barents Seas" published by NRL (Washington, D.C.). The actual chart measures 4 by 3 feet depicting finely grained contours. GENBOT uses bathymetry data files of this area to generate the matrix grids used by ECTRACE.

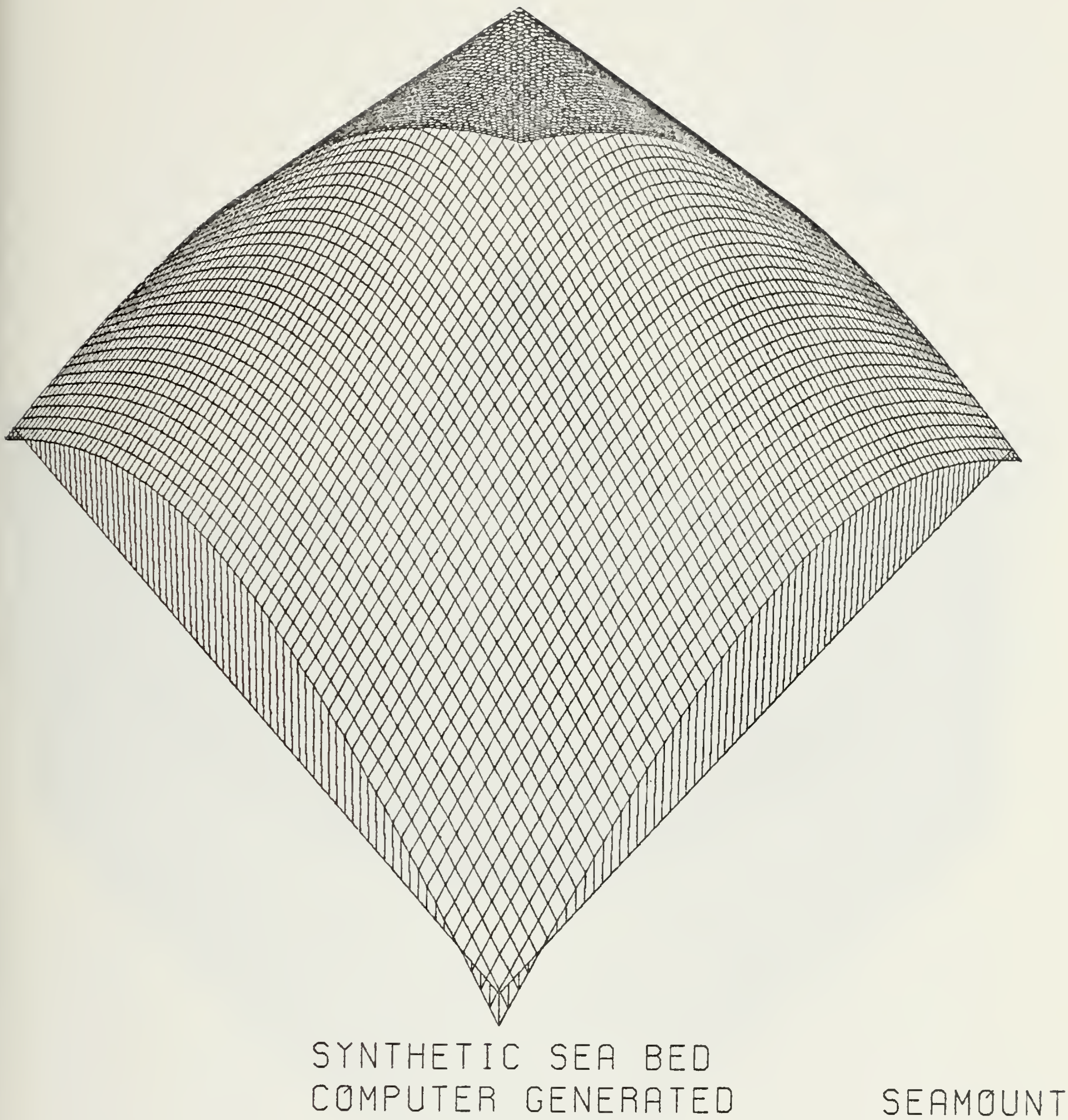


FIGURE 9. A perspective surface plot produced by G3DP of a conical seamount generated by SYNGEN.

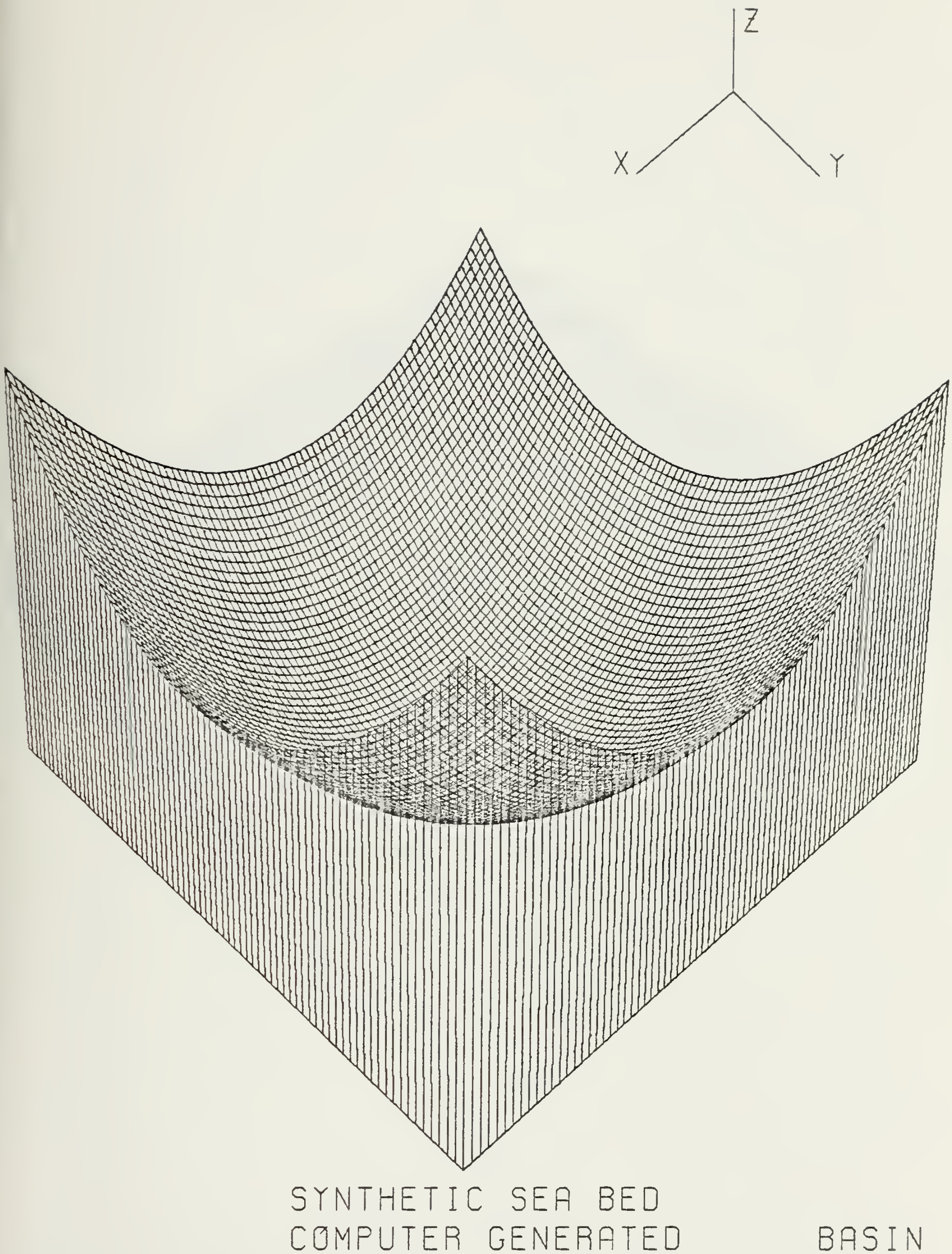


FIGURE 10. A perspective surface plot of a synthetic parabolic basin produced by SYNGEN.

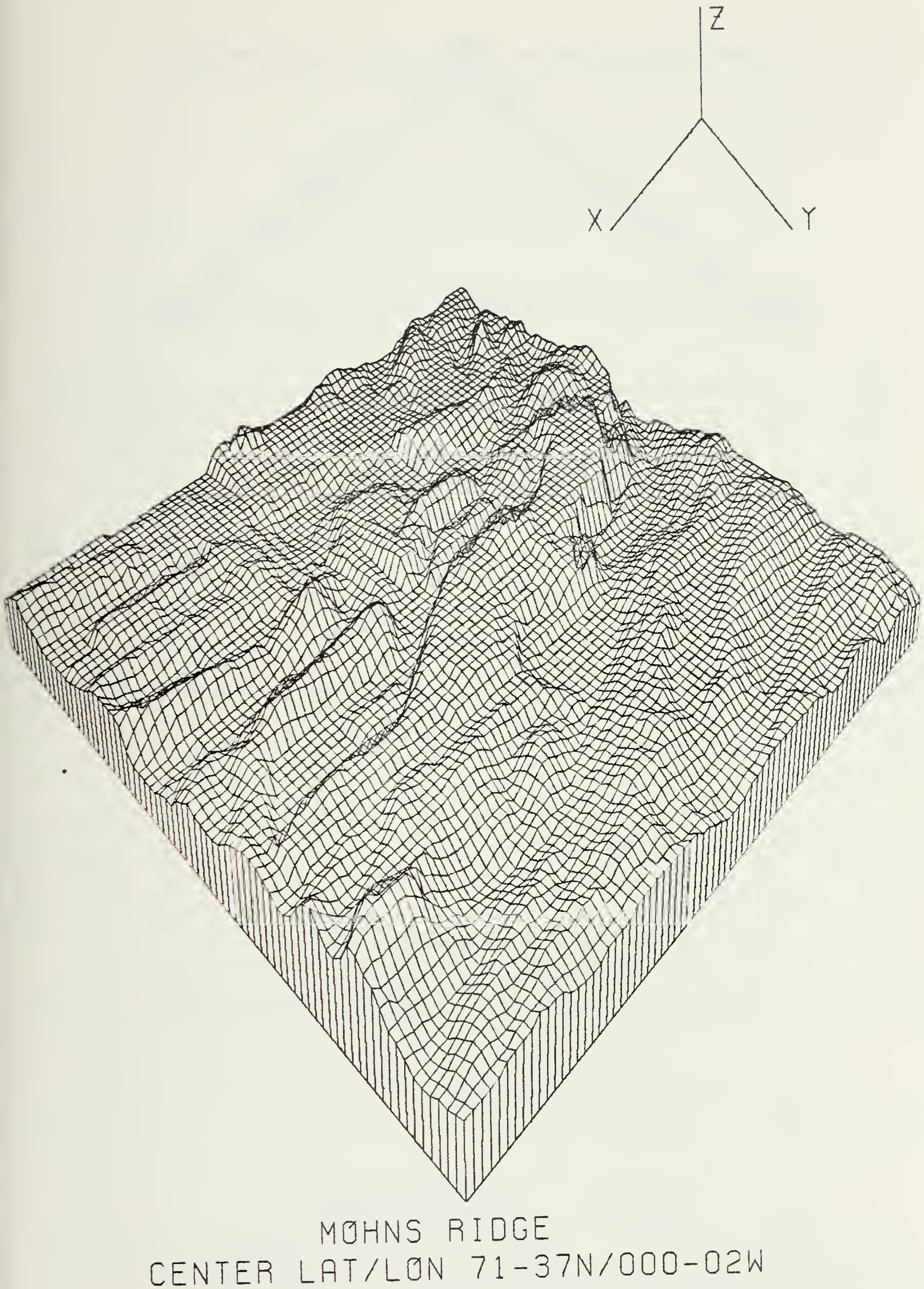


FIGURE 11. A perspective surface plot produced by G3DP of a depth matrix representing a portion of the Mohns Ridge, see Fig. 2.

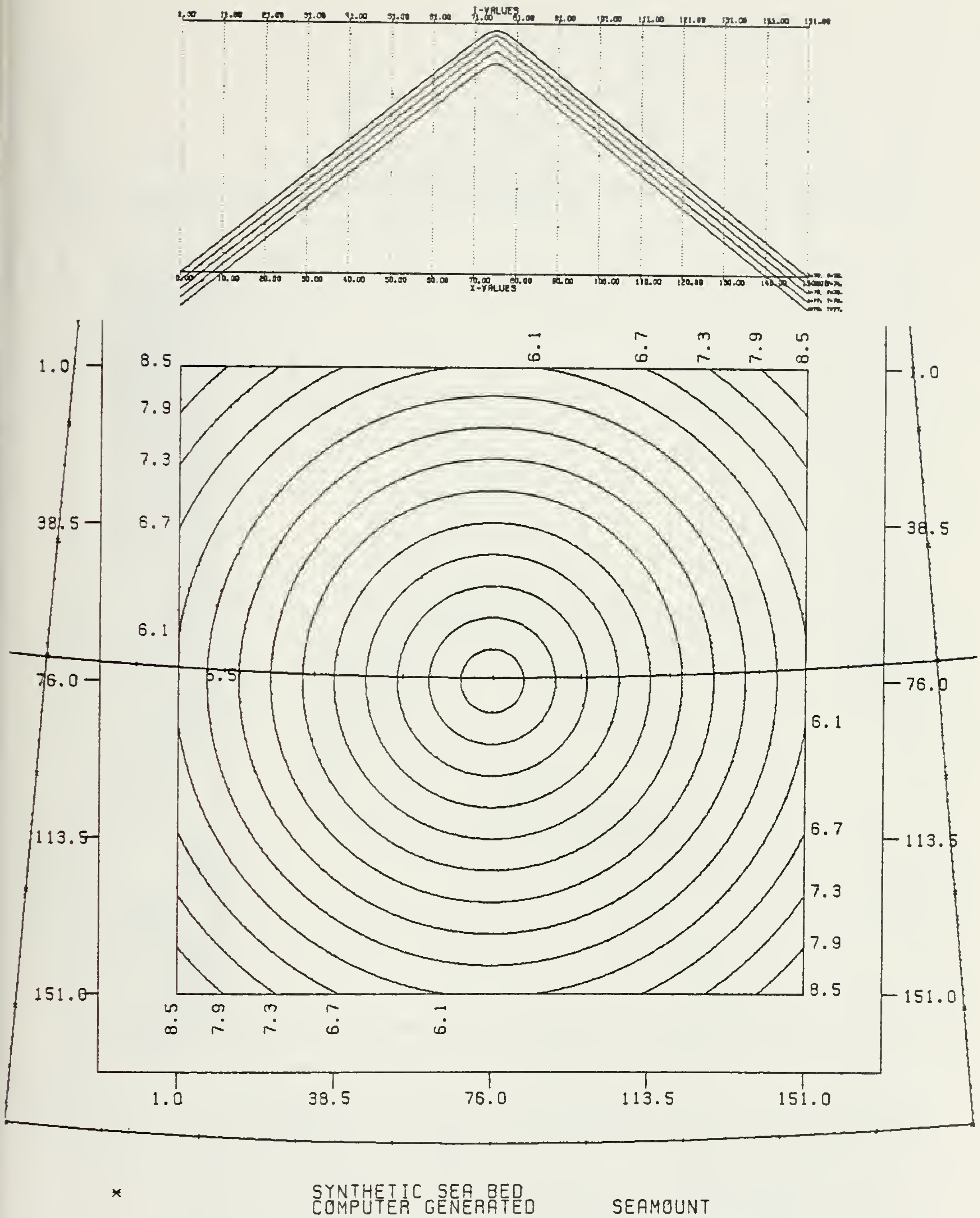


FIGURE 12. A graphical analysis performed by GRDCHK of a depth matrix representing a conical seamount. The upper portion is a series of displaced plots of selected columns of the matrix. The lower portion is a contour plot. The seamount was created by SYNGEN, see Fig. 9.

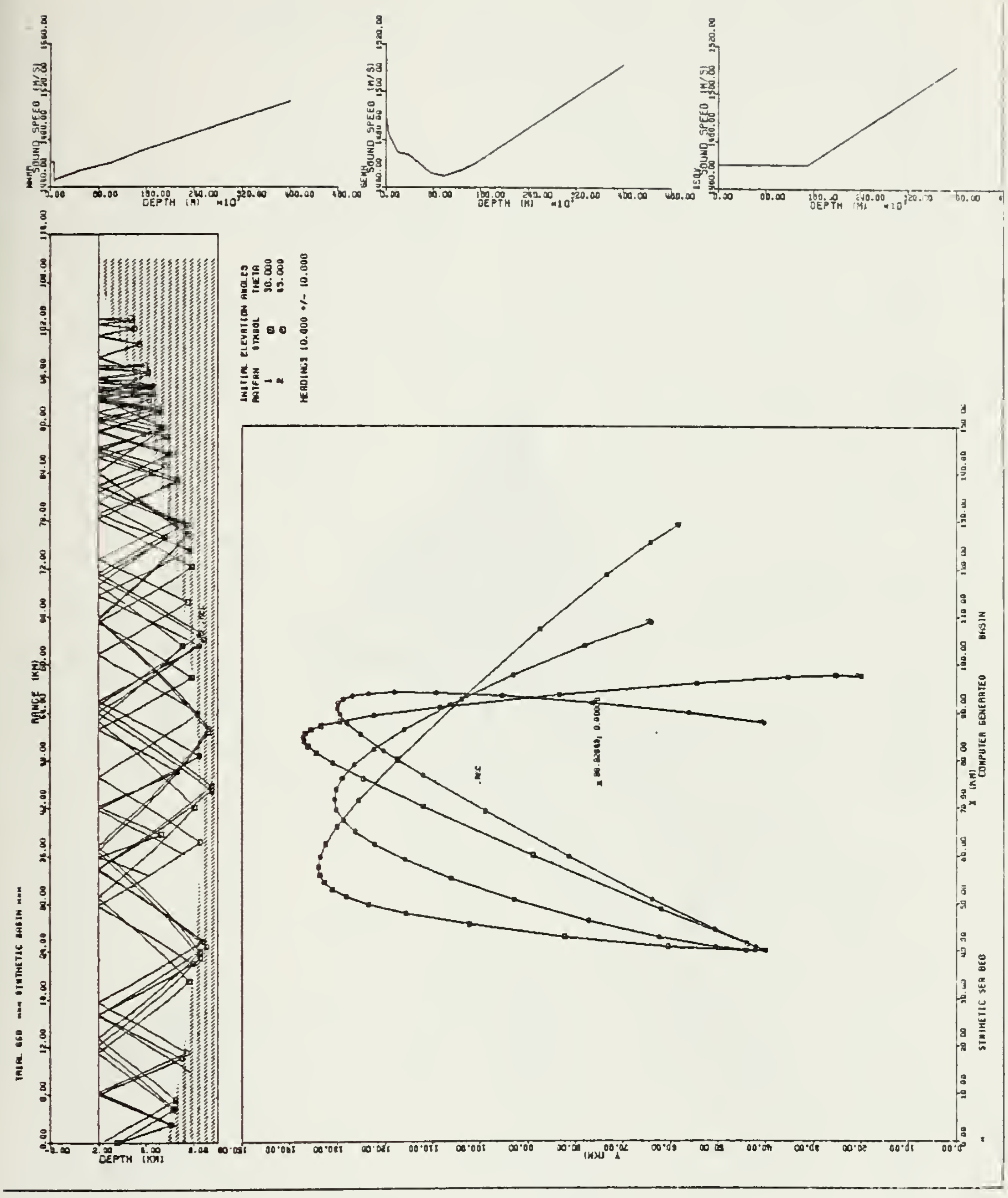


FIGURE 13. An ECTRACE plot of two ray fans in a synthetic parabolic basin (FIGURE 10) generated by SYNGEN. The ray fans have initial elevation angle of 30 and 45 degrees (positive angle downward). Each ray fan contains two rays with initial headings 360 and 020 degrees, respectively. The plot was terminated after each ray had accumulated 23 bottom bounces.

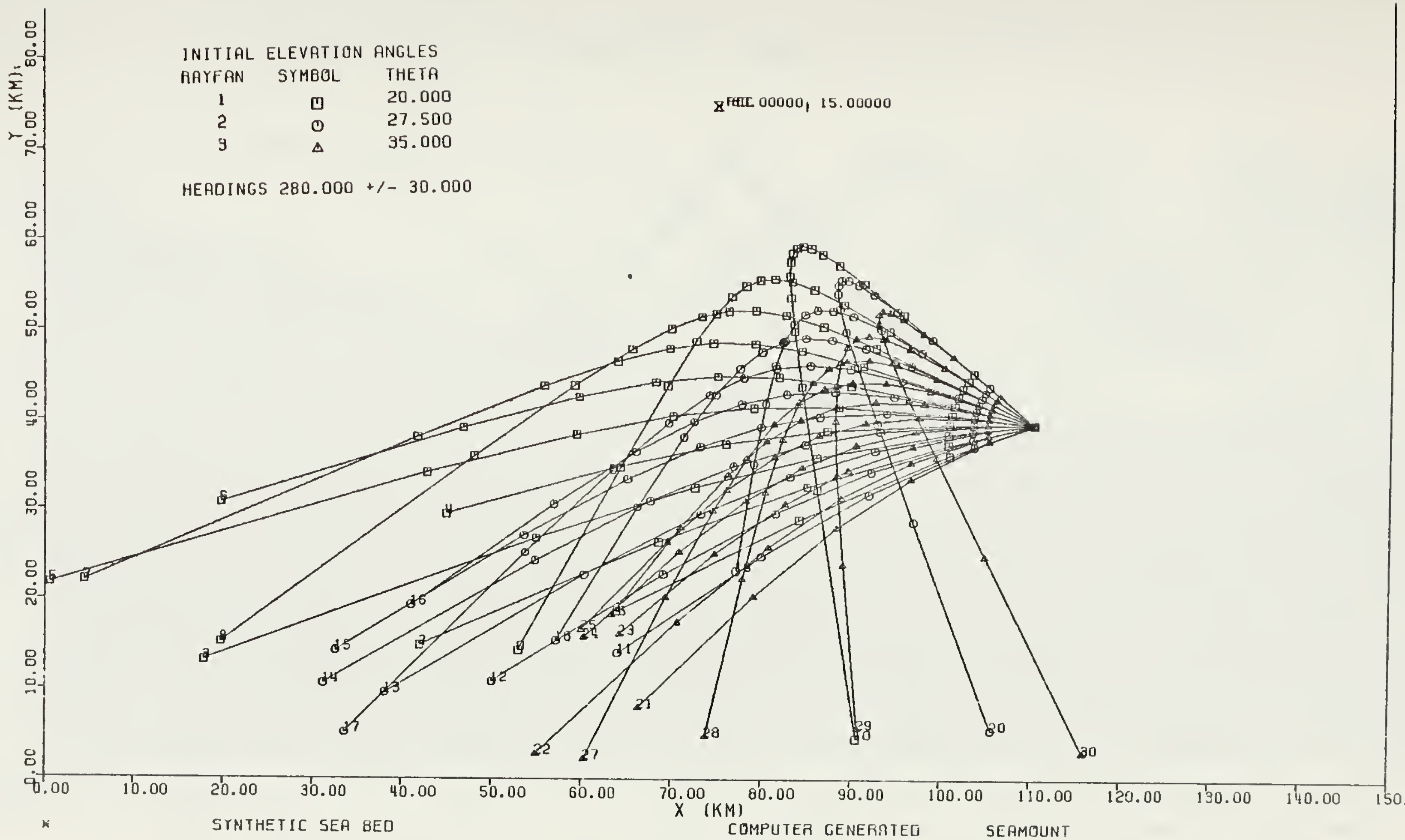


FIGURE 14. An ECTRACE horizontal product of three ray fans contacting a synthetic seamount, see Figs. 9 and 12.

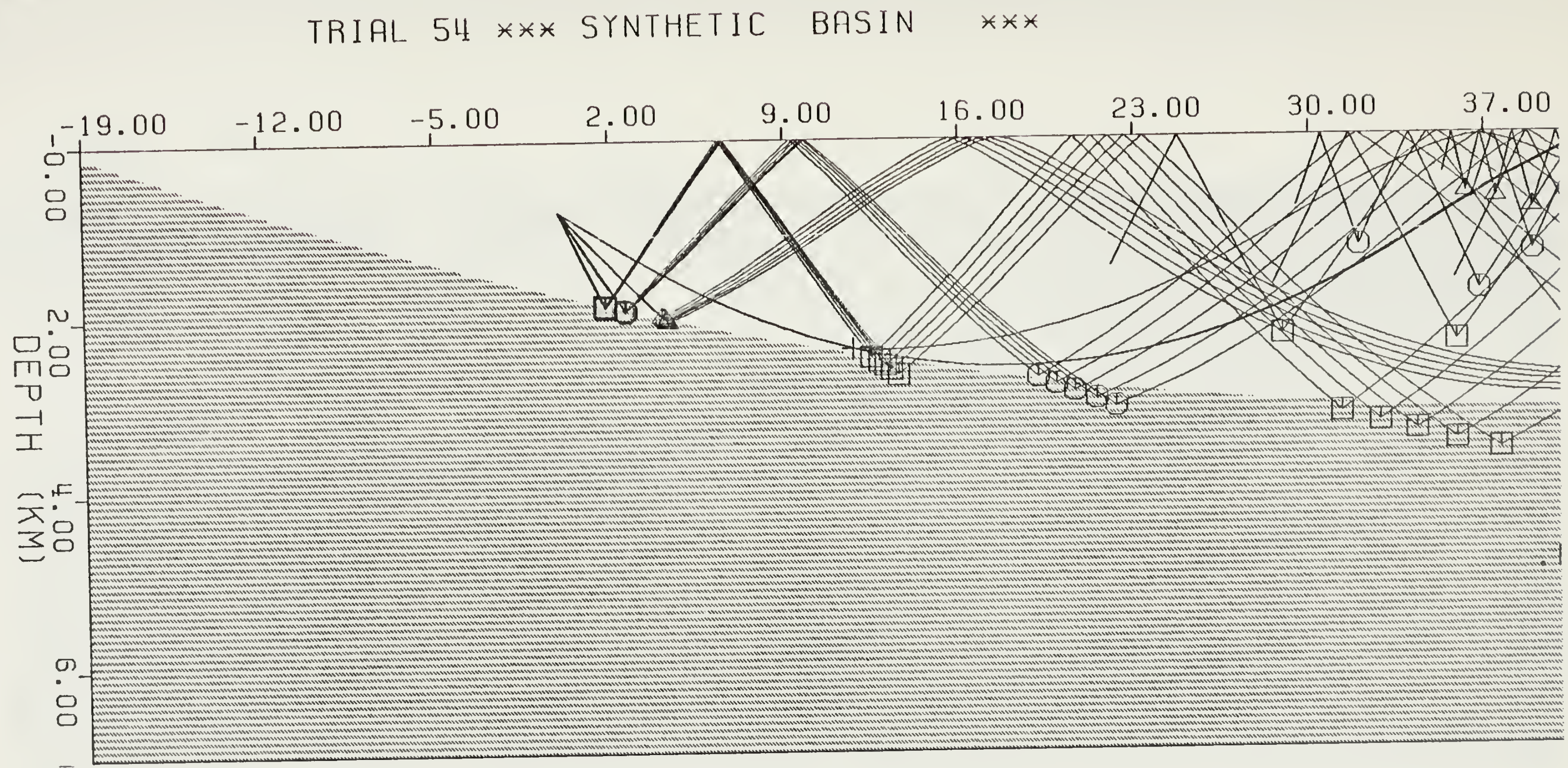
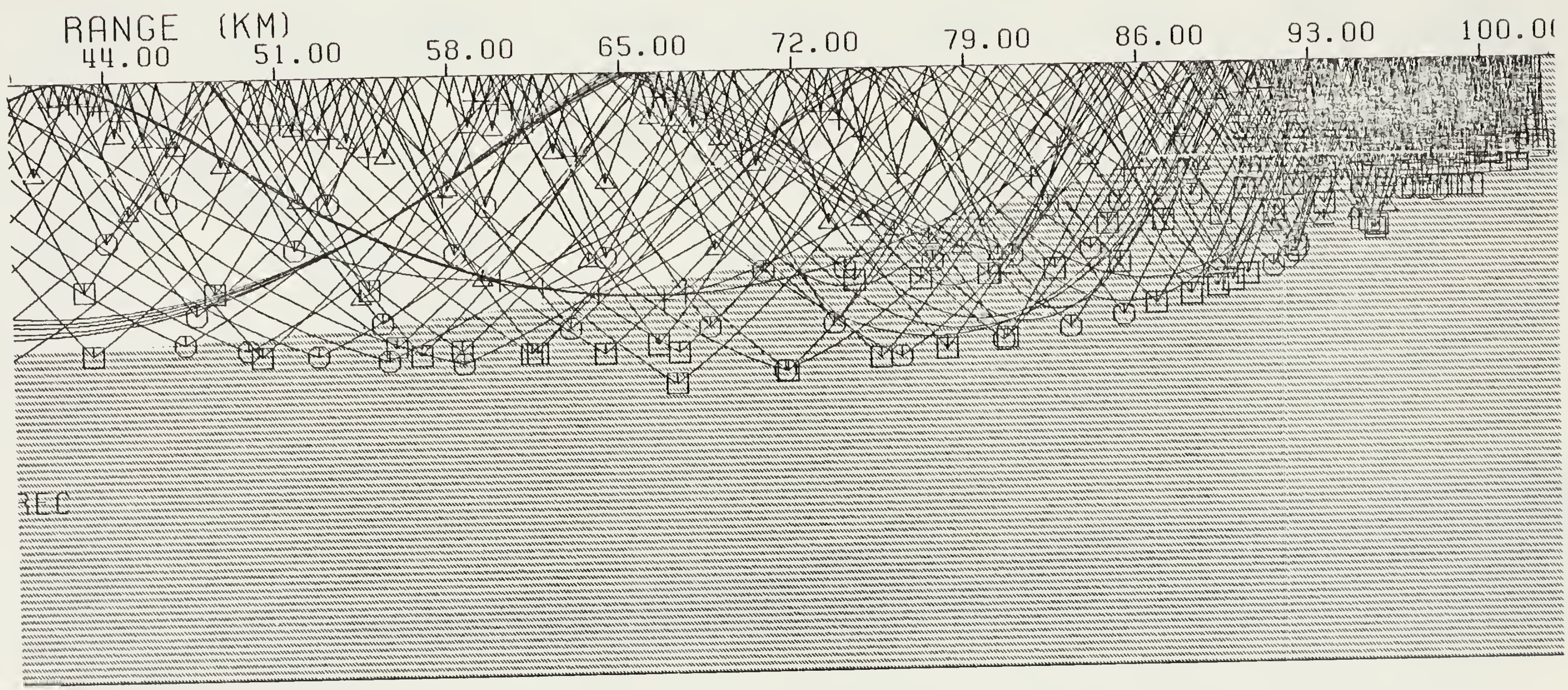


FIGURE 15. ECTRACE TRIAL 54. There are 3 ray fans of 5 rays per fan. The ray fan depicted with the square symbol, can be seen first to curve clockwise away from the bathymetry slice, and then reverse and curve counterclockwise. Eventually, the rays can be seen to appear to approach the source. Since the bottom profile is taken only along the mean heading of the ray fan, most bounce symbols will not lie on the profile surface.

(FIGURE 15 continued)



continued FIGURE 15.

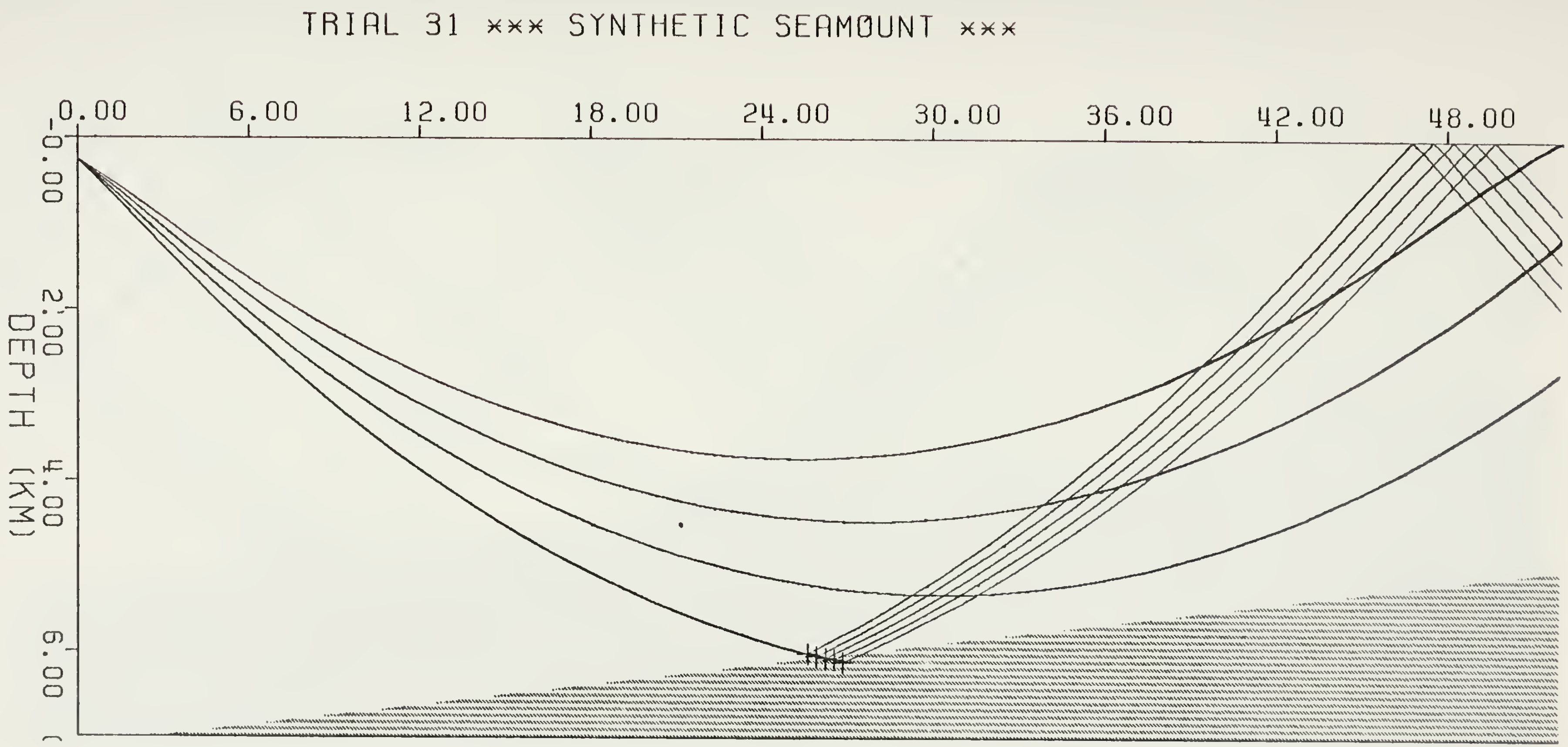
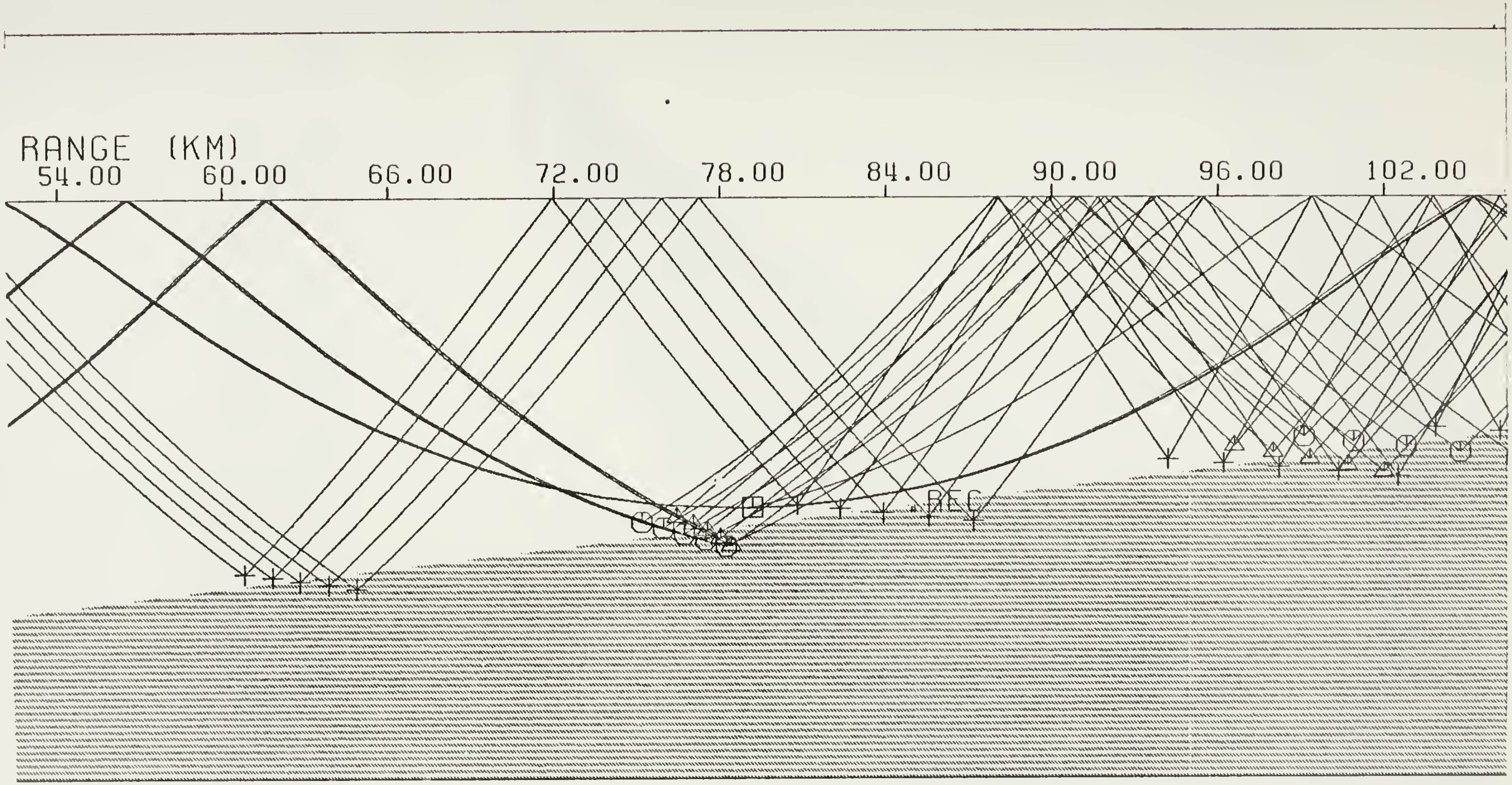


FIGURE 16. The vertical plot of an ECTRACE run over a synthetic grid representing a seamount. The axis of the ray fan was aimed at the center of the seamount. Rays in a single narrow ray fan are nearly indistinguishable until the first bottom contact is made.

(FIGURE 16 continued)



continued FIGURE 16

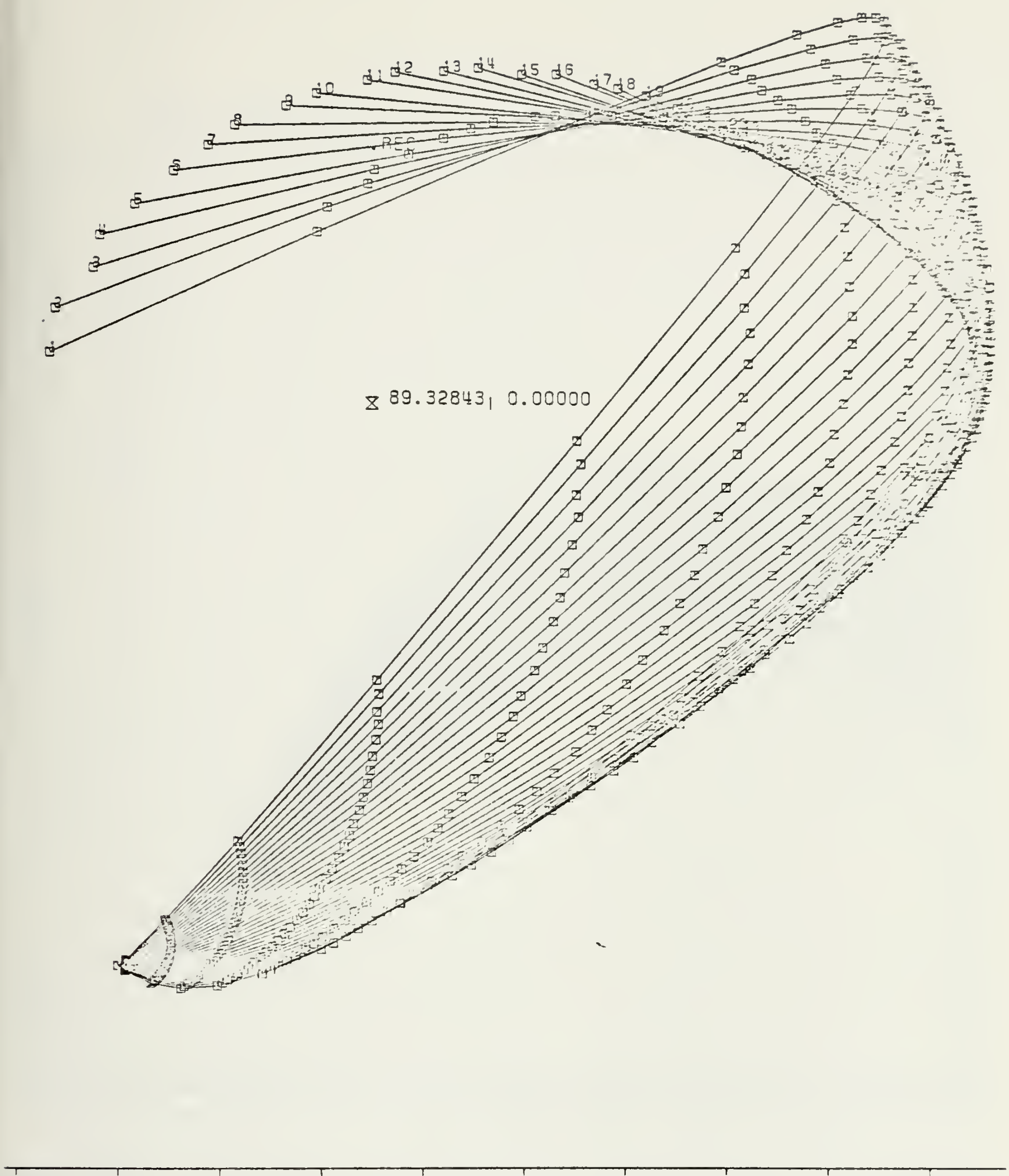


FIGURE 17. An ECTRACE horizontal plot product of a single wide ray fan in a synthetic basin. All rays share the same bottom contact symbol. The number at the last symbol identifies the ray's order in the printed and punched history.

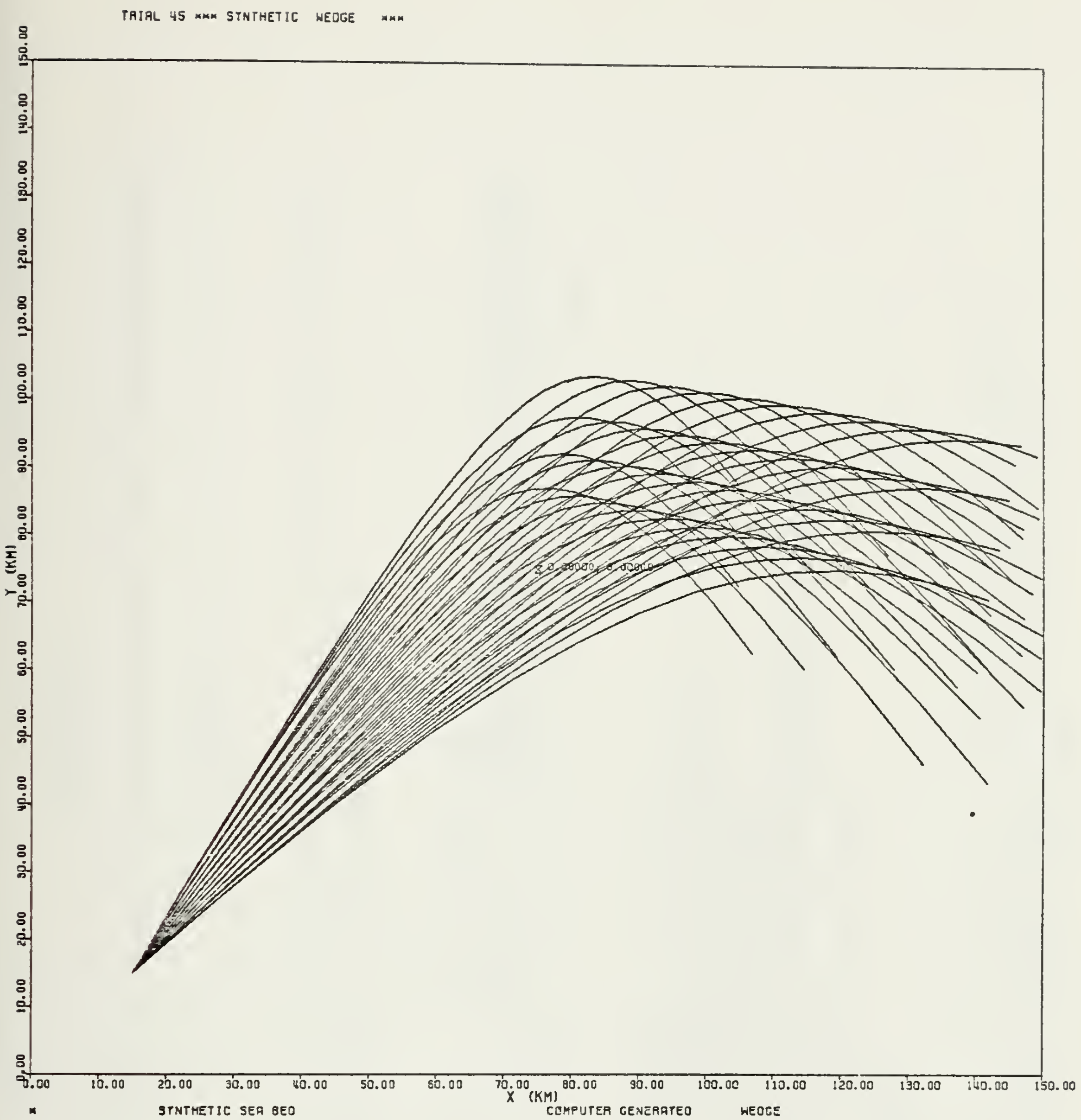


FIGURE 18. A horizontal plot produced by ECCOM of four ray fans propagating over a six degree wedge. The punched output of three ECTRACE runs was combined and then every other ray data history card sequence was removed. There are a total of 40 rays depicted varying in initial elevation angles from 20 to 26 degrees in two degree increments. The initial headings vary from 31 to 49 in two degree increments. The main purpose of ECCOM is to display the horizontal curvature effect of a large number of rays.

```
C
C
C
C                                   APPENDIX H
C
C
C*****************************************************************
C
C
C                                     GENBOT
C
C
C
C
C
C*****************************************************************
C
C//GENBOT   JOB   (2648,0354,IX83),'ERAZO',TIME=20
//   EXEC   FORTCLGW,REGION.GO=250K
//FORT.SYSIN DD *
C          PROGRAM:         GENBOT
C          SOURCE/DATE:     L. R. ERAZO, DEC 79
C          PURPOSE:         READS BATHYMETRY CONTOURS FROM MOBIL OIL CO.
C          DATA FILE AND GENERATES A SQUARE MATRIX OF DEPTH VALUES LYING
C          IN A DESIGNATED REGION OR PRODUCES A CONTOUR PLOT FROM DATA.
C
         SUBROUTINE GEODST MAY BE FOUND IN GRDCHK APPENDIX LISTNNG
C
C
      IMPLICIT REAL*8(B-H),REAL*8(O-W)
      DIMENSION TTL(12)
       DIMENSION AX(5000),AY(5000),AZ(5000),AT(5000),AZB(10)
       DIMENSION NAME(8),GLAT(4),GLON(4)
      INTEGER*2 KEYX(5000),KEYY(5000)
      LOGICAL LEND,LPLOT,LSTP,LGRID
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLON
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,APDV
      DATA NAMET/'NAME'/
      DINT(DPARG) = AINT(SNGL(DPARG))
      ONE=1.
      PI=DATAN(ONE)*4
      DETRA=PI/180.
      KPTT=0
      LSTP=.FALSE.
      IFILE=0
C        USER INPUT GRID PARAMETERS: TITLE, LAT/LONG OF CENTER,
C        SPACING IN KM (RESOLUTION) OF DEPTH VALUES, RADIUS
C        OF AREA. ECHO ON GRID FILE.
      READ 5000,TTL,CNLAT,CNLON,DHN,ARAD
```

```
 5000 FORMAT(6A8/6A8/4F10.5)
      READ 5010,NFILES,LEN
 5010  FORMAT(16I5)
      LGRID=LEN.GT.0
      LEN=IABS(LEN)
      LPLOT=LEN.GE.4
      IF(.NOT.LGRID) GO TO 1
      PRINT 6000,TTL,CNLAT,CNLON,NFILES
 6000 FORMAT(10X,'GRID FILE CALLED ',2(T40,6A8/),10X,'WILL BE CONSTRUCTE
     1D AROUND LAT=',F10.5,' LON=',F10.5/10X,'USING ',I5,' FILES.'/)
      ARADMX=111.12*(89.-DABS(CNLAT))
      ARAD=AMIN1(ARAD,ARADMX)
      IF(ARAD.LT.7.5) GO TO 900
    1 IMAX=IFIX(2*SNGL(ARAD/DHN))+1
      JMAX=IMAX
      XMAX=(IMAX-1)*DHN
      YMAX=XMAX
      XMIN=0.
      YMIN=0.
      XRG=XMAX
      YRG=YMAX
      APDV=AMAX1(XRG,YRG)/15.
      CNX=ARAD
      CNY=ARAD
      CALL CORNER(GLATMN,GLATMX,GLONMN,GLONMX,IMAX,JMAX)
C         PLOT GEOGRAPHIC BOUNDARY LINES.
      DLT=1.
      DLTD=.25
      ABL=DMAX1(DABS(GLATMX),DABS(GLATMN))
      DLN=20.
      IF(ABL.LT.85.) DLN=DLN/2
      IF(ABL.LT.80.) DLN=DLN/2
      DLND=1.
      IF(ABL.GT.75.) GO TO 310
      DLN=1.
      DLND=DLTD
  310 SLON=GLONMN-DLN
      ELON=GLONMX+DLN
      SLAT=GLATMN-DLT
      ELAT=GLATMX+DLT
      IF(SLON.LT.CNLON-90.) SLON=SLON+DLN
      IF(ELON.GT.CNLON+90.) ELON=ELON+DLN
      SLAT=DMAX1(SLAT,DBLE(-89.))
      ELAT=DMIN1(ELAT,DBLE(+89.))
      IF(.NOT.LPLOT) GO TO 9
      AFACT=LEN/15.
      ASIZE=AFACT*21.
      CALL WINDOW(0.,ASIZE,0.,ASIZE)
```

```
      CALL PLOTS(0,0,0)
      CALL FACTOR(AFACT)
      CALL PLOT(0.,0.,-3)
      CALL PLOT(20.99,0.,2)
      CALL PLOT(20.99,20.99,2)
      CALL PLOT(0.,20.99,2)
      CALL PLOT(0.,0.,2)
      CALL PLOT(3.,3.,-3)
      CALL SYMBOL(1.,-2.4,.12,TTL,0.,96)
C         DRAW LATITUDE LINE
      DJ=DLNO
      GI=DINT(SLAT)
      GJ1=DINT(SLON)
      IF(DLN.NE.1.) GJ1=10*DINT(GJ1/10.)
      GJ2=ELON
      IF(GJ1.GE.GJ2) GO TO 50
  340 CALL GEOPLT(GI,GJ1,GJ2,DJ,AX,AY,1)
      GI=GI+DLT
      IF(GI.LE.ELAT) GO TO 340
C         DRAW LONGITUDE LINE
      DJ=DLTD
      GI=GJ1
      GJ1=DINT(SLAT)
      GJ2=ELAT
  350 CALL GEOPLT(GI,GJ1,GJ2,DJ,AX,AY,2)
      GI=GI+DLN
      IF(GI.LE.ELON) GO TO 350
      CALL NUMBER(13.,-2.4,.1,SNGL(DLTD),0.,2)
      CALL SYMBOL(999.,999.,.1,' DEG PER +, ',0.,12)
      CALL NUMBER(999.,999.,.1,SNGL(DLND),0.,2)
      CALL SYMBOL(999.,999.,.1,' DEG PER X',0.,10)
C         BEGIN NEW FILE.
    9 IFILE=IFILE+1
      ICON=0
      KPTF=0
      NPTF=0
      READ (1,5010) LATOR,LONOR,NCON
      LATE=LATOR+10
      LONE=LONOR+10
      PRINT 1900,IFILE,LATOR,LATE,LONOR,LONE,NCON
 1900 FORMAT(//10X,'FILE ',I2,' COVERS LAT ',I3,' TO ',I3,', LON ',
     1I3,' TO ',I3,' WITH ',I5,' CONTOUR LINES. '//T9,'CONTOUR LINE',T23
     2,'DEPTH (KM)',T36,'LABEL',T47,'NPTS',T57,'KPTS')
C         THE NEXT RECORD IS EITHER A LINE LABEL OR THE HEADER RECORD
C         OF A NEW CONTOUR LINE.
   40  READ (1,1400,END=100) (NAME(I),I=1,3),DLAT,DLON,INDEP,METER
 1400  FORMAT(A3,A4,A4,7X,2F10.5,2X,I4,1X,A2)
       IF(NAME(2).EQ.NAMET) GO TO 85
```

```
C          PROCESS A NEW CONTOUR LINE.
      IPLTST=0
      KPTS=0
      NPTS=0
      LEND=.FALSE.
       ICON=ICON+1
      DDD=INDEF/10.
C          BEGIN READING LATS/LONS ALONG CONTOUR LINE.  EACH RECORD
C          HAS FOUR LAT/LON PAIRS.
   45  READ(1,1500) (GLAT(I),GLON(I),I=1,4)
 1500  FORMAT(8F10.5)
       KE=4
       DO 50 K=1,KE
      LEND=GLAT(K).GT.90..OR.GLAT(K).LT.-90.
       IF(LEND) GO TO 60
   50  CONTINUE
C          MORE LINE DATA FOLLOWS.  BUT FIRST:
      GO TO 65
C          END OF CONTOUR LINE. SET KE TO NUMBER OF POINTS IN RECORD.
   60  KE=K-1
       IF(KE.LT.1) GO TO 80
C          CHECK RELEVANCE OF POINTS.
   65  DO 70 K=1,KE
       IF(GLAT(K).LT.SLAT.OR.GLAT(K).GT.ELAT)  GO TO 70
       IF(GLON(K).LT.SLON.OR.GLON(K).GT.ELON) GO TO 70
C          FIND POINT'S POSITION ON GRID.
       CALL GEODST(CNLAT,CNLON,GLAT(K),GLON(K),HS,HF,DIST,-1)
      KPTS=KPTS+1
      KPTT=KPTT+1
      IF(KPTT.LT.4998) GO TO 66
      PRINT 6555
 6555 FORMAT(10X,'*** DIMENSION LIMIT REACHED ***')
      LSTP=.TRUE.
      LEND=.TRUE.
   66 IF(IPLTST.EQ.0) IPLTST=KPTT
      IPLTEN=KPTT
      AY(IPLTEN)=DIST*DCOS(HS)+CNY
      AX(IPLTEN)=DIST*DSIN(HS)+CNX
      AZ(IPLTEN)=DDD
   70 CONTINUE
      NPTS=NPTS+KE
      IF(.NOT.LEND) GO TO 45
   80 PRINT 1600,ICON,DDD,NAME(1),NAME(2),NAME(3),NPTS,KPTS
 1600 FORMAT(10X,I10,F10.2,1X,A3,2A4,I8,I10)
      NPTF=NPTF+NPTS
       IF (KPTS.LE.0) GO TO 40
      KPTF=KPTF+KPTS
      IF((.NOT.LPLOT.OR.KPTS.LE.1).AND..NOT.LSTP) GO TO 40
```

```
C          PLOT CONTOUR LINE.
      KPP1=KPTT+1
      KPP2=KPTT+2
      AX(KPP1)=XMIN
      AY(KPP1)=YMIN
      AX(KPP2)=APDV
      AY(KPP2)=APDV
      CALL LINE(AX(IPLTST),AY(IPLTST),KPTS,1,0,0)
      IF(.NOT.LSTP) GO TO 40
C          END OF CONTOUR FILE.
  100 PRINT 1700,KPTF,NPTF,KPTT,ICON
 1700 FORMAT(10X,'NUMBER OF BOTTOM POINTS USED=',I7,' (OUT OF ',I5,'), A
     1CCUM TOTAL =',I7/10X,'NO. OF CONTOUR LINES PROCESSED FROM FILE=',I
     25)
      IF((IFILE.LT.NFILES).AND..NOT.LSTP) GO TO 9
C          SORT DATA FOR GRID CONSTRUCTION.
      IJ=0
      IERT=0
      CALL RAIN1(KPTT,AX,AY,AT,KEYX,KEYY,IER)
      X1=AX(KEYX(1))+30.
      Y1=AY(KEYY(1))+30.
      X2=AX(KEYX(KPTT))-30.
      Y2=AY(KEYY(KPTT))-30.
      I=1
      J=1
      IF(XMIN.GE.X1.AND.XMAX.LE.X2) GO TO 401
      IHAF=IMAX/2
      DO 400 I=2,IHAF
      XMIN=XMIN+DHN
      XMAX=XMAX-DHN
      IF(XMIN.LE.XMAX) GO TO 401
      IF(XMIN.GE.X1.AND.XMAX.LE.X2) GO TO 401
  400 CONTINUE
  401 IMAX=IMAX-2*(I-1)
      CNX=DHN*FLOAT(IMAX-1)/2.
      IF(YMIN.GE.Y1.AND.YMAX.LE.Y2) GO TO 411
      JHAF=JMAX/2
      DO 410 J=2,JHAF
      YMIN=YMIN+DHN
      YMAX=YMAX-DHN
      IF(YMIN.LE.YMAX) GO TO 411
      IF(YMIN.GE.Y1.AND.YMAX.LE.Y2) GO TO 411
  410 CONTINUE
  411 JMAX=JMAX-2*(J-1)
      CNY=DHN*FLOAT(JMAX-1)/2.
      IF(.NOT.LPLOT) GO TO 210
C          PLOT GRID BOUNDARY LINES.
      XST=XMIN/APDV
```

```
      YST=YMIN/APDV
      XEND=XMAX/APDV
      YEND=YMAX/APDV
      YVAL=2*CNY/APDV
      XVAL=2*CNX/APDV
      CALL AXIS(XST,YST,'Y (KM)',6,YVAL,90.,YMIN,APDV)
      CALL PLOT(XST,YEND,2)
      CALL AXIS(XST,YST,'X (KM)',-6,XVAL,0.,XMIN,APDV)
      CALL PLOT(XEND,YST,2)
      CALL PLOT(XEND,YEND,2)
      CALL PLOT(XST,YEND,2)
      CALL PLOT(0.,0.,999)
  210 CALL CORNER(GLATMN,GLATMX,GLONMN,GLONMX,IMAX,JMAX)
      ARAD=AMIN1(CNX,CNY)
      IF(ARAD.GE.7.5) GO TO 215
  900 PRINT 6900,ARAD
 6900 FORMAT(10X,'SEMI-AXIS LENGTH OF ',F6.2,' KM TOO SMALL.'/10X,
     1   'NO GRID WILL BE PRODUCED.')
      STOP
  215 IF(.NOT.LGRID) STOP
C         COMMENCE GRID CONSTRUCTION.
      WRITE(2,5030)TTL,CNLAT,CNLON,DHN,IMAX,JMAX
      NREC=3
 5030 FORMAT(6A8/6A8/3F10.5,2I5)
      DO 200 J=1,JMAX
      AYJ=DHN*(J-1)+YMIN
      DO 201 I=1,IMAX
      IJ=IJ+1
      AXI=DHN*(I-1)+XMIN
      CALL RAIN2 (KPTT,AX,AY,AZ,KEYX,KEYY,AXI,AYJ,AZK,IER)
      IF(IER.EQ.0) GO TO 202
      IERT=IERT+1
C     PRINT 6030, AXI,AYJ,I,J,AZK,IER
 6030 FORMAT(10X,'X=',F8.4,' Y=',F8.4,' ZB(',I3,',',I3,')=',F8.4,' IER='
     1,I5)
      IF(IERT.GE.10000) GO TO 205
  202 AZB(IJ)=AZK
      IF(IJ.LT.10) GO TO 201
      WRITE(2,1020) AZB
      NREC=NREC+1
      IJ=0
  201 CONTINUE
  200 CONTINUE
      IF(IJ.EQ.0) GO TO 205
      WRITE(2,1020) (AZB(II),II=1,IJ)
      NREC=NREC+1
 1020 FORMAT(10F8.4)
  205 PRINT 6010, NREC,IERT
```

```
 6010 FORMAT(/10X,'GRID CONSTRUCTION COMPLETED.',I5,' RECORDS IN FILE.'/
     110X,I5,' POINTS IN QUESTION.')
      ENDFILE 2
      STOP
C         DRAW LINE VALUES ON LINE.
   85 IF(.NOT.LPLOT) GO TO 40
      IF(DLAT.LT.SLAT.OR.DLAT.GT.ELAT) GO TO 40
      IF(DLON.LT.SLON.OR.DLON.GT.ELON) GO TO 40
      CALL GEODST(CNLAT,CNLON,DLAT,DLON,HS,HF,DIST,-1)
      AXS=(DIST*DSIN(HS)+CNX)/APDV
      AYS=(DIST*DCOS(HS)+CNY)/APDV
      CALL SYMBOL(AXS,AYS,.1,METER,0.,2)
      GO TO 40
      END
      SUBROUTINE CORNER(GLATMN,GLATMX,GLONMN,GLONMX,IMAX,JMAX)
      IMPLICIT REAL*8(B-H),REAL*8(O-W)
      DIMENSION TTL(12)
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLON
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,APDV
      DIST=DSQRT(CNX**2+CNY**2)
C         CALCULATE LAT/LON OF GRID CORNERS. C1 IS SW AND C2 IS SE.
      HS=DATAN2(CNX,CNY)
      CALL GEODST(CNLAT,CNLON,C3Y,C3X,HS,HF,DIST,1)
      HS=DATAN2(CNX,-CNY)
      CALL GEODST(CNLAT,CNLON,C2Y,C2X,HS,HF,DIST,1)
      C1Y=C2Y
      C4Y=C3Y
      DLON=C3X-CNLON
      C4X=CNLON-DLON
      DLON=C2X-CNLON
      C1X=CNLON-DLON
C         CALCULATE BOUNDARY LATS/LONS THAT FORM SPHERICAL RECTANGLE
C         WHICH INCLUDES GRID.
      IF(CNLAT.GE.0.) GO TO 5
      GLATMX=C3Y
      GLONMN=C1X
      GLONMX=C2X
      CALL GEODST(CNLAT,CNLON,GLATMN,CNLON,180.,HFD,CNY,2)
      GO TO 6
    5 GLATMN=C2Y
      GLONMN=C4X
      GLONMX=C3X
      CALL GEODST(CNLAT,CNLON,GLATMX,CNLON,0.,HF,CNY,1)
    6 PRINT 6010,XMIN,YMIN,C1Y,C1X,XMAX,YMIN,C2Y,C2X,XMAX,YMAX,
     1   C3Y,C3X,XMIN,YMAX,C4Y,C4X,GLONMN,GLONMX,GLATMN,GLATMX,IMAX,JMAX
 6010 FORMAT(/T11,'VALUES AT GRID CORNERS:'/T21,'X',T31,'Y',T40,
     1   'LAT',T50,'LON'/4(17X,4F10.5/)/T14,'MIN LON',3X,'MAX LON',
     2   3X,'MIN LAT',3X,'MAX LAT',3X,'IMAX',3X,'JMAX'/T11,4F10.5,2I7/)
```

```
      RETURN
      END
      SUBROUTINE GEOPLT(GI,GJ1,GJ2,DJ,X,Y,LLL)
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      DIMENSION X(183),Y(183),TTL(12)
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLON
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,APDV
      INTEG=LLL+2
      IF(LLL.NE.1) GO TO 10
C        PLOT LATITUDE LINE GI FROM LONGITUDE GJ1 TO GJ2
      PLAT=GI
      PLON=GJ1
      GLAT=GI
      GLON=GJ2
      DLAT=0.
      DLON=DJ
      GO TO 20
C        PLOT LONGITUDE LINE GI FROM LATITUDE GJ1 TO GJ2
   10 PLAT=GJ1
      PLON=GI
      GLAT=GJ2
      GLON=GI
      DLAT=DJ
      DLON=0.
   20 NPTS=0
   21 CALL GEODST(CNLAT,CNLON,PLAT,PLON,HS,HF,DIST,-1)
      IF(PLAT.GT.GLAT.OR.PLON.GT.GLON) GO TO 30
      NPTS=NPTS+1
      X(NPTS)=DIST*DSIN(HS)+CNX
      Y(NPTS)=DIST*DCOS(HS)+CNY
      PLAT=PLAT+DLAT
      PLON=PLON+DLON
      GO TO 21
   30 NPP1=NPTS+1
      NPP2=NPTS+2
      X(NPP1)=XMIN
      X(NPP2)=APDV
      Y(NPP1)=YMIN
      Y(NPP2)=APDV
      CALL LINE(X,Y,NPTS,1,1,INTEG)
      RETURN
      END
      SUBROUTINE GEODST(SLAT,SLON,FLAT,FLON,AZ1,AZ2,DKM,MODE)
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      LOGICAL LDEG
      DATA EPS/.081819356/,ERAVO/6356.75023/
      FUNK(SPSI)=DTAN(DATAN(EPS*SPSI/R1ME2)/2.)**2
      ONE=1.
```

```
      HAPI=DATAN(ONE)*2
      PI=HAPI*2
      TUPI=2*PI
      DETRA=PI/180.
      EPS2=EPS**2
      R1ME2=DSQRT(ONE-EPS2)
      IF(SLAT.EQ.0.) SLAT=1.E-5
      PHI1=SLAT*DETRA
      XLN1=SLON*DETRA
      IF(DABS(SLAT).LT.90.) GO TO 1
      TPSI1=DSIGN(DBLE(5.E9),SLAT)
      GO TO 2
    1 TPSI1=DTAN(PHI1)*R1ME2
    2 PSI1=DATAN(TPSI1)
      LDEG=IABS(MODE).NE.1
      DKM=DABS(DKM)
      IF(MODE) 200,100,100
C        FORWARD SOLUTION.
  100 ALFA1=AZ1
      IF(LDEG) ALFA1=AZ1*DETRA
      IF(ALFA1.EQ.TUPI) ALFA1=0.
      IF(ALFA1.LT.0.) ALFA1=ALFA1+TUPI
      IF(ALFA1.LE.PI) GO TO 101
      DKM=-DKM
      ALFA1=ALFA1-PI
  101 SALFA1=DSIN(ALFA1)
      CPSIM=DCOS(PSI1)*SALFA1
      CALFA1=DCOS(ALFA1)
      SIG1=DATAN2(-CALFA1,TPSI1)
      XLB1=DATAN2(-CALFA1,DSIN(PSI1)*SALFA1)
      SPSIM=DSIN(DARCOS(CPSIM))
      XK=FUNK(SPSIM)
      TWSIG1=2*SIG1
      TWS1P=TWSIG1+XK*(DSIN(TWSIG1)-XK*(DSIN(2*TWSIG1)/8.))
      DSP=(1.-XK)*DKM/BRAVO
      TWSP=TWS1P+DSP
      DSIG=DSP+XK*(XK*5*DSIN(2*DSP)*DCOS(2*TWSP)/8.-DCOS(TWSP)*DSIN(DSP)
     1)
      SIG2=SIG1+DSIG
      SIG=(2*SIG1+DSIG)/2.
      XLB2=DATAN2(DSIN(SIG2),DCOS(SIG2)*CPSIM)
      DLB=XLB2-XLB1
      SPSI2=DCOS(SIG2)*SPSIM
      PSI2=DARSIN(SPSI2)
      PHI2=DATAN(DTAN(PSI2)/R1ME2)
      XLN2=XLN1+DLB-CPSIM*(DSIG*(1.-R1ME2)-EPS2*XK*(DSIN(2*DSIG)*DCOS(4*
     1PSI2)*XK/(4*DSIN(DSIG)*DCOS(2*SIG))-DSIG*(1.+XK/2.))*CPSIM/4.)
      FLAT=PHI2/DETRA
```

```
      FLCN=XLN2/DETRA
C        TO GET AZ2:
      GO TO 30
C        INVERSE SOLUTION.
  200 XLN2=FLON*DETRA
      DXLN=XLN2-XLN1
      IF(DXLN.GT.PI) DXLN=TWPI-DXLN
      DXLN=DSIGN(DMAX1(DABS(DXLN),DBLE(2.E-4)),DXLN)
      HDLB1=DXLN/2.
      PHI2=FLAT*DETRA
      TPSI2=DTAN(PHI2)*R1ME2
      PSI2=DATAN(TPSI2)
   10 TEMP=DATAN2((TPSI1-TPSI2)/DTAN(HDLB1),TPSI1+TPSI2)
      XLB1=TEMP-HDLB1
      XLB2=TEMP+HDLB1
      CLB1=DCOS(XLB1)
      PSIM=DATAN(TPSI1/CLB1)
      CPSIM=DCOS(PSIM)
      SIG1=DATAN2(CPSIM*DSIN(XLB1),CLB1)
      SIG2=DATAN2(CPSIM*DSIN(XLB2),DCOS(XLB2))
      DSIG=SIG2-SIG1
      SIG=(SIG2+SIG1)/2.
      XK=FLNK(DSIN(PSIM))
      TERM=XK*DCOS(2*SIG)*DSIN(DSIG)
      HDLB2=(DXLN+EPS2*CPSIM*(DSIG*(2./(R1ME2+1.)-XK/2.)-TERM/2.)/2.)/2.
      IF(DABS(HDLB2-HDLB1).LT.1.E-7) GO TO 20
      HDLB1=HDLB2
      GO TO 10
   20 DKM=BRAVO*(DSIG+TERM-(XK**2)*DSIN(2*DSIG)*DCOS(4*SIG)/8.)/(1.-XK)
      ALFA1=DATAN2(ONE,-DTAN(XLB1)*DSIN(PSI1))
      AZ1=ALFA1
      IF(DKM.LT.0) AZ1=AZ1+PI
      IF(LDEG) AZ1=AZ1/DETRA
   30 IF(DABS(DABS(XLB2)-HAFI).GE.1.E-7) GO TO 35
      ALFA2=ALFA1
      GO TO 36
   35 ALFA2=DATAN2(ONE,DTAN(XLB2)*SPSI2)
   36 AZ2=ALFA2
      IF(LDEG) AZ2=AZ2/DETRA
      DKM=DABS(DKM)
      RETURN
      END
      SUBROUTINE RAIN1 (N,X,Y,Z,KEYX,KEYY,IER)                          RAIN1940
C   THIS ROUTINE PUTS ASCENDING INTEGERS INTO KEYX(N) AND KEYY(N)      RAIN1950
C   SO THAT RAIN3 CAN ARRANGE THE X'S AND Y'S IN ASCENDING ORDER.      RAIN1960
C                                                                      RAIN1970
      DIMENSION  X(N),Y(N),Z(N)                                        RAIN1980
      INTEGER*2  KEYX(N),KEYY(N)                                       RAIN1990
```

```
      IER = 0                                                           RAIN2000
      IF (N .LE. 32767) GO TO 1                                         RAIN2010
      IER = 1                                                           RAIN2020
      RETURN                                                            RAIN2030
    1 DO 2  I=1,N                                                       RAIN2040
      KEYX(I) = I                                                       RAIN2050
    2 Z(I) = X(I)                                                       RAIN2060
      CALL RAIN3 (Z,KEYX,N)                                             RAIN2070
      DO 3 I=1,N                                                        RAIN2080
      KEYY(I) = I                                                       RAIN2090
    3 Z(I) = Y(I)                                                       RAIN2100
      CALL RAIN3 (Z,KEYY,N)                                             RAIN2110
      RETURN                                                            RAIN2120
      END                                                               RAIN2130
      SUBROUTINE RAIN3 (A,KEY,N)                                        RAIN2140
C    THIS ROUTINE RETURNS TWO ARRAYS, KEYX(N) AND KEYY(N), OF INTEGERS  RAIN2150
C    WHICH SHOW ASSENDING VALUES OF X(N),Y(N).                          RAIN2160
C                                                                       RAIN2170
      INTEGER*2  KEY(N),IT                                              RAIN2180
      DIMENSION  A(N)                                                   RAIN2190
      M1 = 1                                                            RAIN2200
    1 M1 = M1*2                                                         RAIN2210
      IF (M1 .LE. N)  GO TO 1                                           RAIN2220
      M1 = M1/2-1                                                       RAIN2230
      MM = MAX0(M1/2,1)                                                 RAIN2240
      GO TO 3                                                           RAIN2250
    2 MM = MM/2                                                         RAIN2260
      IF (MM .LE. 0) GO TO 6                                            RAIN2270
    3 K = N-MM                                                          RAIN2280
      DO 5 J=1,K                                                        RAIN2290
      II=J                                                              RAIN2300
    4 IM=II+MM                                                          RAIN2310
      IF (A(IM) .GE. A(II))  GO TO 5                                    RAIN2320
      TEMP = A(II)                                                      RAIN2330
      IT = KEY(II)                                                      RAIN2340
      A(II) = A(IM)                                                     RAIN2350
      KEY(II) = KEY(IM)                                                 RAIN2360
      A(IM) = TEMP                                                      RAIN2370
      KEY(IM) = IT                                                      RAIN2380
      II = II-MM                                                        RAIN2390
      IF (II .GT.0)   GO TO 4                                           RAIN2400
    5 CONTINUE                                                          RAIN2410
      GO TO 2                                                           RAIN2420
    6 RETURN                                                            RAIN2430
      END                                                               RAIN2440
      SUBROUTINE RAIN2 (N,X,Y,Z,KEYX,KEYY,XI,YI,ZI,IER)
      DIMENSION KEY(4), IFC(12), X(N), Y(N), Z(N)                       RAIN2460
      DIMENSION  AMX(3,3),ENX(3),DIST(12),A(9),NX(4)                    RAIN2470
```

```
      EQUIVALENCE (A(1),AMX(1,1)),(NX(1),N1),(NX(2),N2),(NX(3),N3),
     1 (NX(4),N4)
      INTEGER*2 KEYX(N),KEYY(N),NUMOQD(4,3),KEYADS(4),KBOUND(4),
     1 NDIST(12)
      DATA JX/-1/,JY/1/,SIZER/15.0/,JSTCEL/0/
C
C     IER IS THE ERROR RETURN; NN3 IS THE MINIMUM NUMBER OF POINTS ALLOW-
C     ED BEFORE INTERPOLATION; N1,N2,N3,N4 COUNT THE NUMBER OF POINTS
C     IN EACH OF THE FOUR QUADRANTS SURROUNDING XI,YI;  AND NUMOQD IS A
C     4X3 MATRIX WHICH STORES THE KEY OF EACH POINT USED FOR INTERPOLATION
C     ACCORDING TO THE QUADRANT IT IS IN, AND HOLDS A MAXIMUM OF 3
C     POINTS PER QUADRANT.
      IER=0
      NN3=3
      N1=0
      N2=0
      N3=0
      N4=0
C
      DO 55 J=1,3
      EMX(J) = 0.0
C
      DO 1 I=1,3
    1 AMX(I,J) = 0.0
      DO 55 I=1,4
   55 NUMOQD(I,J)=0
C
C     DEFINE MIN. AND MAX. X AND Y;  DEFINE RANGE X AND RANGE Y;
C     DEFINE DANGER AREAS WITHIN 5% OF THE BOUNDARIES, AND BOUNDARIES OF
C     EXTRAPOLATION WHICH EXTEND A MAXIMUM OF 5% BEYOND THE RANGE
C     OF POINTS.
      IF (JX.NE.-1) GO TO 8
      JX = 1
      XMIN = X(KEYX(1))
      XMAX = X(KEYX(N))
      YMIN = Y(KEYY(1))
      YMAX = Y(KEYY(N))
      RANGEX = ABS (XMAX - XMIN)
      RANGEY = ABS (YMAX - YMIN)
      AUXX = .05 * RANGEX
      AUXY = .05 * RANGEY
      DNGRXL = XMIN + AUXX
      DNGRXR = XMAX - AUXX
      DNGRYB = YMIN + AUXY
      DNGRYT = YMAX - AUXY
      BNDXL = XMIN - AUXX
      BNDXR = XMAX + AUXX
```

```
      BNDYB = YMIN - AUXY
      BNDYT = YMAX + AUXY
C
C     KNUM IS A COUNTER FOR THE NUMBER OF POINTS WE SEARCH THROUGH TO
C     FIND POINTS IN THE CELL.
C     USE N/10 IN PLACE OF N/20 IN THE FOLLOWING STATEMENT WHEN GRID
C     TYPE DATA IS USED.  FOR EXAMPLE:
C       KNUM=MAXO(3,N/10)
        KNUM=MAXO(3,N/20)
        KBOUND(1)=0
        KBOUND(2)=-N-1
        KBOUND(3)=0
        KBOUND(4)=-N-1
        NPQ=3
        TOL = 1.E-7 * (ABS(XMAX) +ABS(XMIN) +ABS(YMAX) + ABS(YMIN))
C
C     NOW CHECK TO SEE IF XI OR YI ARE WITHIN 5% OF THE RANGE DISTANCE
C     FROM THE BOUNDARY LINES DEFINED BY THE RANGE OF POINTS.
C     NQDFIL IS THE NUMBER OF QUADRANTS WE WANT TO FILL.
      8 NQDFIL=4
        IF (XI .LE .DNGRXL .OR. XI .GE. DNGRXR)  GO TO 11
      9 IF (XI .LT. XMIN .OR. XI .GT. XMAX)  GO TO 6
     75 IF (YI .LE. DNGRYB .OR. YI .GT. DNGRYT)  GO TO 10
     76 IF (YI .LT. YMIN .OR. YI .GT. YMAX)  GO TO 7
        GO TO 2
     10 NQDFIL = MAXO (NQDFIL-2,1)
        IER=5
        GO TO 76
     11 NQDFIL = 2
        IER=4
        GO TO 9
C
C     IF THE POINT IS BEYOND OUR RANGE, SEE IF IT IS WITHIN 5% OVER THE
C     BOUNDARY.  IF SO, WE CAN EXTRAPOLATE, IF NOT THEN ERROR RETURN.
      6 IF (XI .LT. BNDXL .OR. XI .GT. BNDXR)  GO TO 66
C
C      IER =7     XI IS BEYOND THE BOUNDARY, BUT 5% OR LESS.
        IER = 7
        GO TO 75
C
C      IER =2     XI IS OUT OF RANGE
     66 IER = 2
        RETURN
      7 IF (YI .LT. BNDYB .OR. YI .GT.BNDYT)  GO TO 67
C
C      IER =8     YI IS BEYOND THE BOUNDARY, BUT 5% OR LESS.
        IER = 8
        GO TO 2
```

```
C
C     IER =3     YI IS OUT OF RANGE
   67 IER = 3
      RETURN
C
C     FIND THE AREA OF OUR X,Y GRID WHERE THE UNKNOWN POINT XI,YI FITS.
    2 IX=KEYX(JX)
      IF (X(IX).LE.XI) GO TO 3
      JX=JX-1
      IF (JX .GE. 1)  GO TO 2
      JX = 1
      GO TO 4
    3 IX=KEYX(JX+1)
      IF (X(IX).GE.XI) GO TO 4
      JX=JX+1
      IF (JX .LE. N)  GO TO 3
      JX = N
      GO TO 4
    4 IY=KEYY(JY)
      IF (Y(IY).LE.YI) GO TO 5
      JY=JY-1
      IF (JY .GE. 1)  GO TO 4
      JY = 1
      GO TO 12
    5 IY=KEYY(JY+1)
      IF (Y(IY).GE.YI) GO TO 12
      JY=JY+1
      IF (JY .LE. N)  GO TO 5
      JY = N
C
C     NPC IS THE NUMBER OF POINTS FOUND IN THE CELL.
   12 NPC=0
      NQDFM1=MAX0(NQDFIL-1,1)
      NUMINC=0
C
C     THIS DEFINES THE CELL SIZE.
   13 SFN=FLOAT(N)/SIZER
C
C     DEFINE THE CELL LENGTH AND HEIGHT AS A FUNCTION OF THE SMALLER OF
C     THE TWO RANGES.
      XR = RANGEX
      IF (RANGEX .GT. RANGEY)  XR = RANGEY
      XR = XR/SFN
      YR = XR
C
C     K COUNTS THE NUMBER OF CYCLES OF 4 POINTS WE SEARCH.
      K = 0
C
```

```
C     NOW WE COUNT CYCLES OF 4 POINTS.
        KEY(1)=KEYX(JX)
        KEYADS(1)=JX
        KEY(2)=KEYX(JX+1)
        KEYADS(2)=JX+1
        KEY(3)=KEYY(JY)
        KEYADS(3)=JY
        KEY(4)=KEYY(JY+1)
        KEYADS(4)=JY+1
C
C     ESTABLISH THE FOUR LINES OF CELL LIMIT, A SQUARE CENTERED ON THE
C     POINT (XI,YI).  FOR NOW, XR = YR.
        XRT=XI+XR
        XLFT=XI-XR
        YTOP=YI+YR
        YBTM=YI-YR
C
C     THIS LOOP CHECKS TO SEE IF THE 4 POINTS ARE IN THE CELL, IF SO,
C     IT PUTS EACH IN IT'S PROPER QUADRANT.
C
   14 DO 21 I=1,4
C
C      DON'T CHECK THE FIRST CELL SIZE FOR REPETITIONS
C       IF JX OR JY AREN'T AT THE BOUNDARIES OF THEIR ARRAYS,
C       OR IF THE CELL SIZE HASN'T BEEN INCREASED.
C       OR IF THERE ARE NO POINTS IN THE CELL...
C
C      KEYADS  IS THE KEY ADDRESS,   KBOUND IS THE BOUNDARY OF X AND Y
C      ARRAYS,   NUMINC  IS THE NUMBER OF CELL SIZE INCREASES.
        IF (KEYADS(I).LE.KBOUND(I)) GO TO 21
        IF (NUMINC.EQ.0.AND.NPC.EQ.0) GO TO 16
C
        DO 15 J=1,NPC
        IF (IPC(J).EQ.KEY(I)) GO TO 21
   15 CONTINUE
C
C
C      SEE IF THE POINTS ARE IN THE CELL.
   16 XVALUE=X(KEY(I))
        YVALUE=Y(KEY(I))
        IF (XVALUE.GT.XRT.OR.XVALUE.LT.XLFT.OR.YVALUE.GT.YTOP.OR.YVALUE.LT
       1.YBTM) GO TO 21
C
C      IF YOU GET THIS FAR, THE POINT IS IN THE CELL,
C      YOU CAN TAKE A COFFEE BREAK NOW.
C
C      PLACE THE POINT IN IT'S PROPER QUADRANT USING A MAXIMUM OF NPQ
C      POINTS PER QUADRANT.  FOR NOW, NPQ = 3.
```

```
C     NUMOCD(I,J) HOLDS THE KEY OF A POINT IN QUADRANT I, AND MAXIMUM J   RAIN4390
C     IS 3 (3 POINTS PER QUADRANT).                                       RAIN4400
        IF (XVALUE.LE.XI) GO TO 18                                        RAIN4410
        IF (YVALUE.LE.YI) GO TO 17                                        RAIN4420
        IF (N1.GE.3) GO TO 21                                             RAIN4430
        NUMOCD(1,N1+1)=KEY(I)                                             RAIN4440
        N1=N1+1                                                           RAIN4450
        GO TO 20                                                          RAIN4460
   17   IF (N4.GE.3) GO TO 21                                             RAIN4470
        NUMOCD(4,N4+1)=KEY(I)                                             RAIN4480
        N4=N4+1                                                           RAIN4490
        GO TO 20                                                          RAIN4500
   18   IF (YVALUE.LE.YI) GO TO 19                                        RAIN4510
        IF (N2.GE.3) GO TO 21                                             RAIN4520
        NUMOCD(2,N2+1)=KEY(I)                                             RAIN4530
        N2=N2+1                                                           RAIN4540
        GO TO 20                                                          RAIN4550
   19   IF (N3.GE.3) GO TO 21                                             RAIN4560
        NUMOCD(3,N3+1)=KEY(I)                                             RAIN4570
        N3=N3+1                                                           RAIN4580
   20   NPC=NPC+1                                                         RAIN4590
        IPC(NPC)=KEY(I)                                                   RAIN4600
   21   CONTINUE                                                          RAIN4610
C                                                                         RAIN4620
C     IF NQDFIL QUADRANTS ARE FILLED, INTERPOLATE.                        RAIN4630
C                                                                         RAIN4640
C     NQWPTS IS THE NUMBER OF QUADRANTS WITH POINTS IN THEM.              RAIN4650
        NQWPTS=0                                                          RAIN4660
C                                                                         RAIN4670
        DO 22 I=1,4                                                       RAIN4680
         IF (NUMOCD(I,1).NE.0) NQWPTS=NQWPTS+1                            RAIN4690
   22   CONTINUE                                                          RAIN4700
C                                                                         RAIN4710
         IF (NQWPTS.GE.NQDFIL.AND.NPC.GE.NN3) GO TO 29                    RAIN4720
   23    K = K+1                                                          RAIN4730
C                                                                         RAIN4740
C     IF K .GE. KNUM  WE'LL INCREASE CELL SIZE.  OTHERWISE,               RAIN4750
C     GET THE NEXT FOUR POINTS, GO BACK TO SEE IF ANY OF THESE ARE IN     RAIN4760
C     THE CELL.                                                           RAIN4770
         IF (K.GE.KNUM) GO TO 28                                          RAIN4780
         KEYADS(1)=MAX0(JX-K,0)                                           RAIN4790
         IF (KEYADS(1).EQ.0) GO TO 24                                     RAIN4800
         KEY(1)=KEYX(KEYADS(1))                                           RAIN4810
   24    KEYADS(2)=MIN0(JX+1+K,N+1)                                       RAIN4820
         IF (KEYADS(2).EQ.N+1) GO TO 25                                   RAIN4830
         KEY(2)=KEYX(KEYADS(2))                                           RAIN4840
   25    KEYADS(2)=-KEYADS(2)                                             RAIN4850
                                                                          RAIN4860
```

```
      KEYADS(3)=MAX0(JX-K,0)
      IF (KEYADS(3).EQ.0) GO TO 26
      KEY(3)=KEYY(KEYADS(3))
   26 KEYADS(4)=MIN0(JX+1+K,N+1)
      IF (KEYADS(4).EQ.N+1) GO TO 27
      KEY(4)=KEYY(KEYADS(4))
   27 KEYADS(4)=-KEYADS(4)
      GO TO 14
C
C     THIS INCREASES OUR CELL SIZE.
C     JSTCEL COUNTS THE NUMBER OF POINTS INTERPOLATED WITHOUT A CELL
C     INCREASE.  WHEN IT REACHES 3, THE CELL SIZE IS DECREASED.
   28 SIZER=1.1*SIZER
      JSTCEL=0
      NUMINC=NUMINC+1
C
C     SETTLE FOR ONE FEWER QUADRANTS AT THIS POINT,
C     OR GO BACK FOR MORE POINTS.
      IF (NOWPTS.LT.NQDRM1.OR.NPC.LT.NN3) GO TO 13
C
C     NOW FIT THE LEAST SQUARES PLANE THROUGH THE POINTS TO CALCULATE
C     A VALUE FOR ZI.
C     IF THERE ARE POINTS IN ALL 4 QUADRANTS GO TO 37, OTHERWISE TAKE
C     THE CLOSEST 3 POINTS, PUT A LEAST SQUARES PLANE THROUGH THEM
C     AND EXTRAPOLATE.
   29 IF (NOWPTS.EQ.4) GO TO 37
C
      DO 30 I=1,NPC
      K2=IPC(I)
   30 DIST(I)=SQRT((XI-X(K2))**2+(YI-Y(K2))**2)
C
C
      DO 31 I=1,NPC
   31 NDIST(I)=I
C
      CALL RAIN3 (DIST,NDIST,NPC)
C
C     PUT THE 3 CLOSEST POINTS INTO A MATRIX.
C
C     LOOPING FROM 3 TO NPC WILL CONTINUE UNTIL A NON-COLINEAR SET OF
C     POINTS IS FOUND.  IF ALL POINTS ARE COLINEAR, MORE POINTS MUST BE
C     SEARCHED.
C
      IHOPE = 3
C
   59 K2=IPC(NDIST(1))
      AMX(1,1) = X(K2)
      AMX(1,2) = Y(K2)
```

```
      AMX(1,3) = 1.0
      BMX(1) = -Z(K2)
C
      K2=IFC(NDIST(2))
      AMX(2,1) = X(K2)
      AMX(2,2) = Y(K2)
      AMX(2,3) = 1.0
      BMX(2) = -Z(K2)
C
      K2=IFC(NDIST(IHOPE))
      AMX(3,1) = X(K2)
      AMX(3,2) = Y(K2)
      AMX(3,3) = 1.0
      BMX(3) = -Z(K2)
      GO TO 58
C
C    IF ALL ELSE FAILS, COME HERE. THE 3 POINTS ARE CO-LINEAR.
C    IF THE POINTS ARE COLINEAR, ANOTHER POINT MUST BE CHOSEN, OR
C    MORE POINTS MUST BE SEARCHED FOR NON-COLINEARITY.
   32 IER=6
      IF (IHOPE .LT. NPC)  GO TO 33
      NN3=NN3+1
      GO TO 23
   33 IHOPE = IHOPE + 1
      IF (IHOPE .EQ. 7)  GO TO 68
      GO TO 59
C
C    IF WE GET HERE, TOO MANY POINTS ARE COLINEAR,  ERROR RETURN.
   68 IER = 9
      RETURN
C
C    IF THERE ARE POINTS IN ALL 4 QUADRANTS, TAKE THE NEAREST ONE IN
C    EACH QUAD. AND FIT A LEAST SQUARES PLANE THROUGH THEM.
C
   37 DO 44 I=1,4
      K2=NUMQD(I,1)
      NQ = NX(I)
      IF (NQ .EQ. 1)  GO TO 39
      TQ = SQRT ((XI-X(K2))**2 + (YI-Y(K2))**2)
C
C    SEARCH FOR THE CLOSEST POINT IN QUADRANT I.
      DO 41 J=2,NQ
      K3 = NUMQD(I,J)
      TPQ = SQRT ((XI-X(K3))**2 + (YI-Y(K3))**2)
      IF (TPQ .GE. TQ)  GO TO 41
      TQ = TPQ
      K2 = K3
   41 CONTINUE
```

```
   39 AMX(1,1) = AMX(1,1) + X(K2)**2                                     RAIN5820
      AMX(2,1) = AMX(2,1) + X(K2)*Y(K2)                                  RAIN5830
      AMX(3,1) = AMX(3,1) + X(K2)                                        RAIN5840
      AMX(2,2) = AMX(2,2) + Y(K2)**2                                     RAIN5850
      AMX(3,2) = AMX(3,2) + Y(K2)                                        RAIN5860
      BMX(1) = BMX(1) - X(K2)*Z(K2)                                      RAIN5870
      BMX(2) = BMX(2) - Y(K2)*Z(K2)                                      RAIN5880
      BMX(3) = BMX(3) - Z(K2)                                            RAIN5890
   44 CONTINUE                                                           RAIN5900
      AMX(1,2) = AMX(2,1)                                                RAIN5910
      AMX(1,3) = AMX(3,1)                                                RAIN5920
      AMX(2,3) = AMX(3,2)                                                RAIN5930
      AMX(3,3) = 4.                                                      RAIN5940
C                                                                        RAIN5950
C    SOLVE THE 3 EQUATIONS & FIND THE PLANE PASSING THROUGH THEM,        RAIN5960
C    THEN EXTRAPOLATE.                                                   RAIN5970
C                                                                        RAIN5980
C    THE FOLLOWING HUNDRED OR SO STATEMENTS SOLVE THE SET OF SIMUL-      RAIN5990
C    TANEOUS EQUATIONS, AMX * X = BMX.                                   RAIN6000
C                                                                        RAIN6010
C        FORWARD SOLUTION                                                RAIN6020
C                                                                        RAIN6030
   58 JJ=-3                                                              RAIN6040
      DO 65 J=1,3                                                        RAIN6050
      JYY=J+1                                                            RAIN6060
      JJ=JJ+4                                                            RAIN6070
      BIGA=0.                                                            RAIN6080
      IT=JJ-J                                                            RAIN6090
      DO 70 I=J,3                                                        RAIN6100
C                                                                        RAIN6110
C        SEARCH FOR MAXIMUM COEFFICIENT IN COLUMN                        RAIN6120
C                                                                        RAIN6130
      IJ=IT+I                                                            RAIN6140
      IF (ABS(BIGA) .GE. ABS(A(IJ)))  GO TO 70                           RAIN6150
      BIGA=A(IJ)                                                         RAIN6160
      IMAX=I                                                             RAIN6170
   70 CONTINUE                                                           RAIN6180
C                                                                        RAIN6190
C        TEST FOR PIVOT LESS THAN TOLERANCE (SINGULAR MATRIX)            RAIN6200
C                                                                        RAIN6210
      IF (ABS(BIGA) .GT. TOL)  GO TO 80                                  RAIN6220
      GO TO 32                                                           RAIN6230
C                                                                        RAIN6240
C        INTERCHANGE ROWS IF NECESSARY                                   RAIN6250
C                                                                        RAIN6260
   80 I1=J+3*(J-2)                                                       RAIN6270
      IT=IMAX-J                                                          RAIN6280
      DO 90 KK=J,3                                                       RAIN6290
```

```
      I1=I1+3
      I2=I1+IT
      SAVE=A(I1)
      A(I1)=A(I2)
      A(I2)=SAVE
C
C        DIVIDE EQUATION BY LEADING COEFFICIENT
C
   90 A(I1)=A(I1)/BIGA
      SAVE=BMX(IMAX)
      BMX(IMAX)=BMX(J)
      BMX(J)=SAVE/BIGA
C
C        ELIMINATE NEXT VARIABLE
C
      IF (J .EQ. 3)  GO TO 110
      IQS=3*(J-1)
      DO 65 IX=JYY,3
      IXJ=IQS+IX
      IT=J-IX
      DO 100 JXX=JYY,3
      IXJX=3*(JXX-1)+IX
      JJX=IXJX+IT
  100 A(IXJX)=A(IXJX)-(A(IXJ)*A(JJX))
   65 BMX(IX)=BMX(IX)-(BMX(J)*A(IXJ))
C
C        BACK SOLUTION
C
  110 DO 120 J=1,2
      IA=9-J
      IB=3-J
      IC=3
      DO 120 KK=1,J
      BMX(IB)=BMX(IB)-A(IA)*BMX(IC)
      IA=IA-3
  120 IC=IC-1
C
C   THIS IS THE ANSWER.
C   NOW WE COMPLETE THE CALCULATION OF ZI.
      ZI = -BMX(1)*XI -BMX(2)*YI -BMX(3)
C
C   IF 3 POINTS ARE CALCULATED WITHOUT A CELL INCREASE, DECREASE IT.
      JSTCEL=JSTCEL+1
      IF (JSTCEL.LT.3) RETURN
      SIZER=SIZER*0.9
      JSTCEL=0
      RETURN
C     DEBUG INIT, SUBCHK, TRACE
```

```
C       AT 2
C       TRACE CN
C       AT 120
C       TRACE CFF
        END                                                                  FAIN677C
C/*
//GO.FT01FCC1  CD  UNIT=3330,VCL=SER=DISK03,DSN=S2648.CONF30,
//      DISP=SHR
//GC.FT01F002  CC  UNIT=3330,VCL=SER=DISK03,DSN=S2648.CONF24,
//      DISP=SHR
//GO.FT02F001 CC UNIT=333C,VOL=SER=DISK01,DISP=SHR,DSN=S2648.GRID2
//GC.SYSIN  CC  *
    EAST LOFOTEN BASIN TO VESTERAALEN RISE
          CENTER LAT/LON 70-04,15-51
   7C.C7C59  15.85C32   1.C       75.
     2     5
C/*
```

```
C
C
C
C                               APPENDIX J
C
C
C*****************************************************************************
C
C
C                                 SYNGEN
C
C
C
C
C*****************************************************************************
C
C//SYNGEN JOB (     ,0354,IX83),'ERAZO',TIME=2
C
C      USER MAY GENERATE A SYNTHETIC SEA BOTTOM. THE FOLLOWING TYPES
C      TYPES ARE AVAILABLE: PARABOLIC BASIN, CONICAL SEAMOUNT, WEDGE,
C      TROUGH, RIDGE, AND UNDULATING BOTTOM.
C
//   EXEC  FORTCLG,REGION.GO=250K
//FORT.SYSIN  DD  *
      REAL*8 TTL(12),DHN,SYN(17)
      DATA SYN/'*        ','         ','SYNTHETI','C SEA BE','D       ',
     13*'        ','COMPUTER',' GENERAT','ED      ','WEDGE   ',
     2'RIDGE    ','TROUGH   ','SEAMOUNT','UNDULAT.','BASIN    '/
      DIMENSION ZB(151,151)
      READ 5000,CNLAT,CNLON,DMIN,DMAX,ITYPE
 5000 FORMAT(4F10.5,I5)
      DIF=DMAX-DMIN
      DO 10 I=1,11
   10 TTL(I)=SYN(I)
      TTL(12)=SYN(ITYPE)
C         RIDGE, TYPE 13
      IF(ITYPE.NE.13) GO TO 12
      IMAX=11
      JMAX=11
      DHN=15.
      DO 90 J=1,11
      DO 90 I=1,11
   90 ZB(I,J)=DMAX
      DO 130 J=1,11
  130 ZB(6,J)=DMIN
      GO TO 999
C         WEDGE, TYPE 12
   12 IF(ITYPE.NE.12) GO TO 15
```

```
      IMAX=11
      JMAX=11
      DHN=15.
      SLOPE=DIF/9.
      DO 20 J=1,11
      DO 120 I=1,10
  120 ZB(I,J)=DMAX-SLOPE*(I-1)
   20 ZB(11,J)=DMIN
      GO TO 999
C         MOUNT, TYPE 15
   15 IF(ITYPE.NE.15) GO TO 16
      IMAX=151
      JMAX=151
      DHN=1.
      SLOPE=DIF/75.
  151 DO 150 J=1,151
      YSQ=(J-76)**2
      DO 150 I=1,151
      XSQ=(I-76)**2
      RSQ=XSQ+YSQ
  150 ZB(I,J)=DMIN+SLOPE*SQRT(RSQ)
      GO TO 999
C         UNDULATION,TYPE 16
   16 IF(ITYPE.NE.16) GO TO 14
      IMAX=151
      JMAX=151
      DHN=1.
      SLOPE=DIF/2.
      DO 160 I=1,151
      DEP=DMIN+SLOPE*(COS((I-76)*0.2094395)+1.)
      DO 160 J=1,151
  160 ZB(I,J)=DEP
      GO TO 999
C         TROUGH, TYPE 14
   14 IF(ITYPE.NE.14) GO TO 17
      IMAX=11
      JMAX=11
      DHN=15.
      DO 40 J=1,11
      DO 40 I=1,11
   40 ZB(I,J)=DMIN
      DO 140 J=1,11
  140 ZB(6,J)=DEP
      GO TO 999
C         BASIN,TYPE 17
   17 IF(ITYPE.NE.17) GO TO 151
      IMAX=151
      JMAX=151
```

```
      DHN=1.
      SLOPE=DIF/5625
      DO 170 J=1,151
      YSQ=(J-76)**2
      DO 170 I=1,151
      XSQ=(I-76)**2
      RSQ=XSQ+YSQ
  170 ZB(I,J)=DMAX-SLOPE*RSQ
  999 WRITE(2,2000) TTL, CNLAT,CNLON,DHN,IMAX,JMAX
 2000 FORMAT(6A8/6A8/3F10.5,2I5)
      WRITE(2,1000) ((ZB(I,J),I=1,IMAX),J=1,JMAX)
 1000 FORMAT(10F8.4)
      ENDFILE 2
      PRINT 6000,TTL,CNLAT,CNLON,IMAX,JMAX,DMIN,DMAX
 6000 FORMAT(10X,'A SYNTHETIC SEA BED TITLED',2(T40,6A8/)/10X,
     1   'WAS CONSTRUCTED AROUND LAT=',F10.5,' LON=',F10.5//10X,
     2   'GRID DIMENSIONS ARE ',I3,' BY ',I3,', DEPTHS RANGE FROM',F6.3,
     3   ' TO ',F6.3,' KM.')
      STOP
      END
C/*
//GO.FT02F001 DD UNIT=3330,VOL=SER=DISK03,DSN=S2648.SYNBED16,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=800),DISP=SHR,
// SPACE=(TRK,(20,5))
//GO.SYSIN DD *
  89.3        0.        1.        5.        12
C/*
```

```
C
C
C
C                                  APPENDIX  K
C
C
C*************************************************************************
C
C
C                                     G3DP
C
C
C
C*************************************************************************
C
C        SUBROUTINE GRDSCT IS CONTAINED IN THIS LISTING
C        IT IS CALLED BY GRDCHK AND SUBROUTINE TRACER
C
C
C//G3DP2   JOB   (    ,0354,1X83),'ERAZO   SMC1704',TIME=2
//   EXEC   FORTCLGW,REGION.GO=300K
//FORT.SYSIN   DD   *
      REAL*8 TTL(12),CNX,CNY,CNLAT,CNLNG,DHN
      COMMON /DIMS/ KMX,KMY
      COMMON /FLCTIT/ TTL,CNX,CNY,CNLAT,CNLNG
      LOGICAL*1 ICN(76,46),LHAF
      DIMENSION ZE(151,91),D(76,46),F(2),SIZE(2),
     1   WK(76,46,3),KX(2000),KY(2000),X(76),Y(46)
C        ABOVE DIMENSIONS SHOULD EXACTLY EQUAL THE FOLLOWING:
C              1.  ZE(KMX,KMY)
C              2.  D(NROW,NCOL), WK(NROW,NCOL,3), X(NROW), Y(NCOL),
C              ICN(NROW,NCOL) SUCH THAT
C              NROW=INT((KMX+1)/2), NCOL=INT((KMY+1)/2)
C              3.  BELOW DECLARATIONS MUST CONFORM.
      DATA F/2*0./
      KMX=151
      KMY=91
      NROW=(KMX+1)/2
      NCOL=(KMY+1)/2
      READ(5,5000) XST,YST,ZEND,DEGUP,DEGDW,LW
 5000 FORMAT(5F8.3,I3)
      LW=MAX0(LW,6)
      LW=MIN0(LW,18)
      SIZE(1)=LW
      SIZE(2)=LW
      CALL GRDSCT(XST,YST,XMAX,YMAX,IMAX,JMAX,DHN,ZB)
      ICT=2*NROW+1
      JCT=2*NCOL+1
```

```
30 DO 10 I=1, NROW
   II=ICT-2*I
   DO 20 J=1, NCOL
   JJ=JCT-2*J
20 D(I,J)=(ZEND-ZB(II,JJ))*3
   X(I)=I
10 Y(I)=I
   CALL PLT3D1(X,NROW,Y,NCOL,D,DEGUP,DEGCCW,F,TTL,SIZE,WK,ICN,
  1   KX,KY,2000,2)
   STOP
   END
C/*
//LINK.USDD DD UNIT=3330,VOL=SER=DISK01,DSN=S2648.ECTLMCD,
// DISP=SHR,LABEL=(,,,IN)
//LINK.SYSIN DD *
 INCLUDE USDD(GRDSCT)
 ENTRY MAIN
C/*
//GO.FT01F001  DD DSN=S2648.GRID2,UNIT=3330,VOL=SER=DISK01,
//   DISP=SHR,LABEL=(,,,IN)
//GO.SYSIN  DD  *
   0.       0.       4.       30.       45.       6
C/*
```

```
C
C
C
C
C                            SUBROUTINE:
C                                       GRDSCT
C
C
C**********************************************************************
C
C
      SUBROUTINE GRDSCT(XST,YST,XMAX,YMAX,IMAX,JMAX,DHN,ZB)
C
C
C        EXTRACTS KMX*KMY SECTION OF LARGER 2D DATA MATRIX.
C        XST,YST MUST BE MULTIPLES OF 10*DHN.
C        IMAX,JMAX MUST BE GE XST+KMX, YST+KMY, RESPECTIVELY.
C
C       GRDSCT IS CALLED BY THE FOLLOWING PROGRAMS:
C                                                 G3DP
C                                                 GRDCHK
C                                                 ECTRACE -CALLED BY TRACE
C
C
      REAL*8 TTL(12),CNX,CNY,CNLAT,CNLNG,DHN
      COMMON /FLCTIT/ TTL,CNX,CNY,CNLAT,CNLNG
      COMMON /DIMS/ KMX,KMY
      DIMENSION ZB(KMX,KMY), D10(10)
      READ(1,1010) TTL,CNLAT,CNLNG,DHN,IMAX,JMAX
 1010 FORMAT(6A8/6A8/3F10.5,2I5)
      CNX=(IMAX-1)*DHN/2.
      CNY=(JMAX-1)*DHN/2.
      IF(KMX.EQ.IMAX.AND.KMY.EQ.JMAX) GO TO 80
C        READ ONLY A SECTION OF INPUT GRID.
      FAR=DHN
      KMX=MINO(KMX,IMAX)
      KMY=MINO(KMY,JMAX)
      XST=AMIN1(XST,(IMAX-KMX)*FAR)
      YST=AMIN1(YST,(JMAX-KMY)*FAR)
      GRFL=10*DHN
      XST=GRFL*IFIX(XST/GRFL)
      YST=GRFL*IFIX(YST/GRFL)
C        NCSS = NO. OF CARDS REPRESENTING COLUMNS (Y-VALUES) SKIPPED
C        NISW = NO. OF ITEMS SKIPPED, WEST (LEFT)
C        KSAD = ITEM THRESHOLD, DUMMY CONTAINS NO NEEDED DATA (SKIP ALL
C           DUMMY)
C        KSND = ITEM THRESHOLD, DUMMY FULL OF NEEDED DATA (SKIP NO
C           DUMMY)
```

```
      NCSS=IMAX*IFIX(YST/GRFL)                                          GFD00520
      NISW=IFIX(XST)                                                    GFD00530
      KSND=NISW+10                                                      GFD00540
      KSAD=KSND+KMX                                                     GFD00550
      J=1                                                               GFD00560
      ITEM=0                                                            GFD00570
      IF(NCSS) 10,10,4                                                  GFD00580
    4 DO 5 ISKIP=1,NCSS                                                 GFD00590
    5 READ(1,1000,ERR=70,END=70)                                        GFD00600
   10 IZB=0                                                             GFD00610
      IL=1                                                              GFD00620
   20 READ(1,1000,ERR=70,END=70) D10                                    GFD00630
 1000 FORMAT(10F8.4)                                                    GFD00640
      ITEM=ITEM+10                                                      GFD00650
   15 IF(ITEM.LE.NISW) GO TO 20                                         GFD00660
      IF(ITEM.GE.KSAD) GO TO 50                                         GFD00670
      IST=ITEM-NISW                                                     GFD00680
      ID=10*IL-IST+1                                                    GFD00690
      IF(ITEM.GT.KSND) ID=1                                             GFD00700
      IB=IZB+1                                                          GFD00710
      DO 30 I=IB,IST                                                    GFD00720
         ZB(I,J)=D10(ID)                                                GFD00730
         IF(I.EQ.KMX) GO TO 60                                          GFD00740
   30    ID=ID+1                                                        GFD00750
      IZB=IST                                                           GFD00760
   40 IL=IL+1                                                           GFD00770
   50 IF(ITEM.LT.IMAX) GO TO 20                                         GFD00780
      ITEM=ITEM-IMAX                                                    GFD00790
      IZB=0                                                             GFD00800
      IL=1                                                              GFD00810
      GO TO 15                                                          GFD00820
   60 J=J+1                                                             GFD00830
      IF(J.GT.JMAX) GO TO 70                                            GFD00840
      IF(J.LE.KMY) GO TO 40                                             GFD00850
C        READ ENTIRE INPUT GRID                                         GFD00860
   80 READ(1,1000) ZB                                                   GFD00870
      J=KMY+1                                                           GFD00880
   70 J=J-1                                                             GFD00890
   90 ITEMS=KMX*J                                                       GFD00900
      XRAN=(KMX-1)*DHN                                                  GFD00910
      YRAN=(J-1)*DHN                                                    GFD00920
      XMAX=XST+XRAN                                                     GFD00930
      YMAX=YST+YRAN                                                     GFD00940
      PRINT 6000,TTL,CNLAT,CNLNG,CNX,CNY,XST,XMAX,YST,YMAX,ITEMS        GFD00950
 6000 FORMAT('0GRID FILE CALLED ',52('*'),2(/T19,'*',1X,6A8,1X,'*')/T19,GFD00960
     152('*')/' HAS BEEN READ. CENTER LAT/LONG IS ',F10.5,'/',F10.5,    GFD00970
     2' AT (',F7.3,',',F7.3,') KM.'/' RANGES ARE ',F6.1,' TO ',F6.1,    GFD00980
     3' KM IN X AND ',F6.1,' TO ',F6.1,' KM IN Y (',I5,' ITEMS).')      GFD00990
```

```
      RETURN                                                            GFC01000
      END                                                               GFC01010
```

```
C
C
C
C                                      APPENDIX L
C
C
C
C*****************************************************************
C
C
C                                        GRDCHK
C
C
C
C
C
C                          SUBROUTINES:
C                                      CORNER
C                                      GEOPLR
C                                      GEODST
C
C
C
C*****************************************************************
C
C//GRDCHK2 JOB (     ,0354,IX83),'ERAZO  SMC 1704',TIME=2
C
//   EXEC  FORTCLGW,REGION.GO=350K
//FORT.SYSIN  DD  *
      IMPLICIT REAL*8(B-H),REAL*8(O-W)
      LOGICAL*1 LTG(3),LECHO,LPLOT,LGEO
C    SET DIMENSIONS: ZB(KMX,KMY), ZV(KMX*KMY+4) AS KMX, KMY DECLARED BELOW.
      DIMENSION ZV(22805),ZB(151,151),AX(183),AY(183),TTL(12)
      COMMON /DIMS/ KMX,KMY
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLON
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,APDV
      DATA LMASK/Z1111/
      DINT(DPARG) = AINT(SNGL(DPARG))
      KMX=151
      KMY=151
C          CHECK ABOVE PARAMETERS WITH DIMENSION REQUIREMENTS.
      READ(5,5000) NL,LEN
      READ(5,5001) ZST,ZEND,XMIN,YMIN
      LECHO=.FALSE.
      READ(5,5000,END=20) IPS,IPE,JPS,JPE
      LECHO=.TRUE.
 5000 FORMAT(16I5)
 5001 FORMAT(4F6.1)
   20 IANL=IABS(NL)
```

```
      ZDV=(ZEND-ZST)/IANL
      CALL GRDSCT(XMIN,YMIN,XMAX,YMAX,IMAX,JMAX,DHN,ZB)
      NROW=MIN0(IPE-IPS+1,KMX)
      NCOL=MIN0(JPE-JPS+1,KMY)
      JPE=JPS+NCOL-1
      IPE=IPS+NROW-1
      XRG=XMAX-XMIN
      YRG=YMAX-YMIN
      LGEC=LEN.GT.0
      LEN=IABS(LEN)
      LPLOT=LEN.GE.4.AND.IANL.LE.100
      CALL PLOTS(0,0,0)
      LEN=MIN0(LEN,19)
      AFCT=LEN/15.
      YOR=(21.-LEN)/2.
      CALL PLOT(2.,YOR,-3)
      AXY=AMAX1(15.,AMAX1(XRG,YRG))
      AR15=AXY/15.
      APDV=AINT(AR15)
      AFACT=APDV/AR15
      AFCT=AFCT*AFACT
      CALL FACTOR(AFCT)
      IF(.NOT.LECHO) GO TO 25
      XPS=XMIN+DHN*(IPS-1)
      XPE=XMIN+DHN*(IPE-1)
      PRINT 6010,IPS,IPE,XPS,XPE,DHN
 6010 FORMAT('0INPUT ARRAY FOLLOWS, BY COLUMNS: ROWS ',I3,' TO ',I3,
     1  ' OF EACH COL. (X=',F5.1,' TO ',F5.1,'); DHN=',F5.1,' KM.'//)
      AIDV=AINT((NROW-1)/15.)
      AIDV=AMAX1(AIDV,1.)
      AXDV=AIDV*DHN
      FVAL=IPS
      XMOVE=0.
      NX=0
      CALL AXIS(0.,0.,'I-VALUES',8,15.,90.,FVAL,AIDV)
      CALL PLOT(0.,15.,2)
      CALL PLOT(0.,0.,-2)
      IRS=IPS-1
      IRE=IPE-1
      DO 70 I=IRS,IRE
   70 ZV(I+1)=I
      ZV(I+2)=0.
      ZV(I+3)=AIDV
      AZDV=ZEND/5.
      IVS=NROW+3
      IVE=IVS+NROW-1
      ZV(IVE+1)=0.
      ZV(IVE+2)=AZDV
```

```
      DO 10 JJ=1,NCOL
         J=JPE-JJ+1
         YKM=YMIN+DHN*(J-1)
         PRINT 6020,J,YKM
 6020    FORMAT(' COLUMN ',I3,', Y=',F5.1)
         PRINT 6030,(ZB(I,J),I=IPS,IPE)
 6030    FORMAT(1X,22F6.3)
         II=IPS-1
         DO 60 I=IVS,IVE
         II=II+1
   60    ZV(I)=ZB(II,J)
         CALL LINE(ZV(IVS),ZV(IPS),NROW,1,0,0)
         CALL WHERE(XNOW,YNOW,AFCT)
         CALL SYMBOL(XNOW,YNOW,.08,'J=',90.,2)
         FPN=J
         CALL NUMBER(999.,999.,.08,FPN,90.,-1)
         CALL SYMBOL(999.,999.,.08,', Y=',90.,4)
         CALL NUMBER(999.,999.,.08,YKM,90.,0)
         XMOVE=XMOVE+.2
         NX=NX+1
         CALL PLOT(.2,0.,-3)
   10    CONTINUE
      XMOVE=XMOVE+5.
      CALL PLOT(5.,0.,-3)
      NY=IFIX(XRG/APDV+.1)
      NX=NX+25
      CALL AXIS(0.,0.,'X-VALUES',-8,15.,90.,XPS,AXDV)
      CALL PLOT(0.,15.,2)
      CALL GRID(-XMOVE,0.,-NX,.2,NY,1.,LMASK)
      IPEN=999
      IF(LPLOT) IPEN=-IPEN
      CALL PLOT(0.,0.,IPEN)
      IF(.NOT.LPLOT) STOP
C              TRANSPOSE ARRAY TO PUT NORTH AT TOP
   25 DO 40 K=1,KMY
         I=KMY+1-K
      DO 40 J=1,KMX
         IJ=KMX*(J-1)+I
   40 ZV(IJ)=ZB(J,K)
      LTG(1)=.TRUE.
      LTG(2)=.FALSE.
      LTG(3)=.NOT.LGEO
      IF(LTG(3)) GO TO 50
C        PLOT GEOGRAPHIC BOUNDARY LINES.
      XOR=LEN+2
      CALL PLOTS(0,0,0)
      CALL PLOT(XOR,YOR,-3)
      CALL SYMBOL(.3,-.5,.08,TTL,90.,48)
```

```
      CALL SYMBOL(.4,-.5,.08,TTL(7),90.,48)
      CALL FACTOR(AFCT)
      CALL CORNER(GLATMN,GLATMX,GLONMN,GLONMX,IMAX,JMAX)
      DLT=1.
      DLTC=.25
      ABL=DMAX1(DABS(GLATMX),DABS(GLATMN))
      DLN=20.
      IF(ABL.LT.85.) DLN=DLN/2
      IF(ABL.LT.80.) DLN=DLN/2
      DLNC=1.
      IF(ABL.GT.75.) GO TO 310
      DLN=1.
      DLNC=.25
  310 SLON=GLONMN-DLN
      ELON=GLONMX+DLN
      SLAT=GLATMN-DLT
      ELAT=GLATMX+DLT
      IF(SLON.LT.CNLON-90.) SLON=SLON+DLN
      IF(ELON.GT.CNLON+90.) ELON=ELON-DLN
      SLAT=DMAX1(SLAT,DBLE(-89.))
      ELAT=DMIN1(ELAT,DBLE(+89.))
C         DRAW LATITUDE LINE
      DJ=DLNC
      GI=DINT(SLAT)
      GJ1=DINT(SLON)
      IF(DLN.NE.1.) GJ1=10*DINT(GJ1/10.)
      GJ2=ELON
  340 CALL GEOPLR(GI,GJ1,GJ2,DJ,AX,AY,1)
      GI=GI+DLT
      IF(GI.LE.ELAT) GO TO 340
C         DRAW LONGITUDE LINE
      DJ=DLTC
      GI=GJ1
      GJ1=DINT(SLAT)
      GJ2=ELAT
      IF(GJ1.GE.GJ2) GO TO 50
  350 CALL GEOPLR(GI,GJ1,GJ2,DJ,AX,AY,2)
      GI=GI+DLN
      IF(GI.LE.ELON) GO TO 350
   50 AX(1)=ZST
      IF(NL.LT.2) GO TO 35
      DO 30 I=2,NL
         IM1=I-1
   30    AX(I)=AX(IM1)+ZCV
   35 IW=IFIX(YRG*AFCT/APCV+.1)
      IH=IFIX(XRG*AFCT/APCV+.1)
      CALL CONTUR(ZV,KMY,KMX,KMY,AX,NL,TTL,LEN,LEN,LTG)
      STOP
```

```
      END
C
C
      SUBROUTINE CORNER(GLATMN,GLATMX,GLONMN,GLONMX,IMAX,JMAX)
C
C
C        CALCULATES GRID CORNER COORDINATES.
      IMPLICIT REAL*8(B-H),REAL*8(O-W)
      DIMENSION TTL(12)
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLON
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,ARCV
      DIST=DSQRT(CNX**2+CNY**2)
C        CALCULATE LAT/LON OF GRID CORNERS. C1 IS SW AND C2 IS SE.
      HS=DATAN2(CNX,CNY)
      CALL GEODST(CNLAT,CNLON,C3Y,C3X,HS,HF,DIST,1)
      HS=DATAN2(CNX,-CNY)
      CALL GEODST(CNLAT,CNLON,C2Y,C2X,HS,HF,DIST,1)
      C1Y=C2Y
      C4Y=C3Y
      DLON=C3X-CNLON
      C4X=CNLON-DLON
      DLON=C2X-CNLON
      C1X=CNLON-DLON
C        CALCULATE BOUNDARY LATS/LONS THAT FORM SPHERICAL RECTANGLE
C        WHICH INCLUDES GRID.
      IF(CNLAT.GE.0.) GO TO 5
      GLATMX=C3Y
      GLONMN=C1X
      GLONMX=C2X
      CALL GEODST(CNLAT,CNLON,GLATMN,CNLON,180.,HFD,CNY,2)
      GO TO 6
    5 GLATMN=C2Y
      GLONMN=C4X
      GLONMX=C3X
      CALL GEODST(CNLAT,CNLON,GLATMX,CNLON,0.,HF,CNY,1)
    6 PRINT 6010,XMIN,YMIN,C1Y,C1X,XMAX,YMIN,C2Y,C2X,XMAX,YMAX,
     1   C3Y,C3X,XMIN,YMAX,C4Y,C4X,GLONMN,GLONMX,GLATMN,GLATMX,IMAX,JMAX
 6010 FORMAT(/T11,'VALUES AT GRID CORNERS:'/T21,'X',T31,'Y',T40,
     1   'LAT',T50,'LON'/4(17X,4F10.5/)/T14,'MIN LON',3X,'MAX LON',
     2   3X,'MIN LAT',3X,'MAX LAT',3X,'IMAX',3X,'JMAX'/T11,4F10.5,2I7/)
      RETURN
      END
C
C
      SUBROUTINE GEOPLR(GI,GJ1,GJ2,DJ,XNEW,YNEW,LLL)
C
C
      IMPLICIT REAL*8(C-H),REAL*8(O-W)
```

```
      DIMENSION TTL(12),XNEW(183),YNEW(183)
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLON
      COMMON /CHART/ XMIN,YMIN,XMAX,YMAX,XRG,YRG,APDV
      INTEG=LLL+2
      IF(LLL.NE.1) GO TO 10
C        PLOT LATITUDE LINE GI FROM LONGITUDE GJ1 TO GJ2
      PLAT=GI
      PLON=GJ1
      GLAT=GI
      GLON=GJ2
      DLAT=0.
      DLON=DJ
      GO TO 20
C        PLOT LONGITUDE LINE GI FROM LATITUDE GJ1 TO GJ2
   10 PLAT=GJ1
      PLON=GI
      GLAT=GJ2
      GLON=GI
      DLAT=DJ
      DLON=0.
   20 NPTS=0
   21 CALL GEODST(CNLAT,CNLON,PLAT,PLON,HS,HF,DIST,-1)
      IF(PLAT.GT.GLAT.OR.PLON.GT.GLON) GO TO 30
      NPTS=NPTS+1
C        MUST BE TRANSPOSED TO LIE ON ROTATED CONTOUR PLOT.
      YNEW(NPTS)=DIST*DSIN(HS)+CNX
      XNEW(NPTS)=YRG-(DIST*DCOS(HS)+CNY)
      PLAT=PLAT+DLAT
      PLON=PLON+DLON
      GO TO 21
   30 NPP1=NPTS+1
      NPP2=NPTS+2
      YNEW(NPP1)=XMIN
      YNEW(NPP2)=APDV
      XNEW(NPP1)=YRG-YMIN
      XNEW(NPP2)=APDV
      CALL LINE(XNEW,YNEW,NPTS,1,1,INTEG)
      RETURN
      END
C
C
      SUBROUTINE GEODST(SLAT,SLON,FLAT,FLON,AZ1,AZ2,DKM,MODE)
C
C
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      LOGICAL LDEG
      DATA EPS/.081819356/,ERAVO/6356.75023/
      FUNK(SPSI)=DTAN(DATAN(EPS*SPSI/R1ME2)/2.)**2
```

```fortran
      ONE=1.
      HAPI=DATAN(ONE)*2
      PI=HAPI*2
      TUPI=2*PI
      DETRA=PI/180.
      EPS2=EPS**2
      R1ME2=DSQRT(ONE-EPS2)
      IF(SLAT.EQ.0.) SLAT=1.E-5
      PHI1=SLAT*DETRA
      XLN1=SLON*DETRA
      IF(DABS(SLAT).LT.90.) GO TO 1
      TPSI1=DSIGN(DBLE(5.E9),SLAT)
      GO TO 2
    1 TPSI1=DTAN(PHI1)*R1ME2
    2 PSI1=DATAN(TPSI1)
      LDEG=IABS(MODE).NE.1
      DKM=DABS(DKM)
      IF(MODE) 200,100,100
C        FORWARD SOLUTION.
  100 ALFA1=AZ1
      IF(LDEG) ALFA1=AZ1*DETRA
      IF(ALFA1.EQ.TUPI) ALFA1=0.
      IF(ALFA1.LT.0.) ALFA1=ALFA1+TUPI
      IF(ALFA1.LE.PI) GO TO 101
      DKM=-DKM
      ALFA1=ALFA1-PI
  101 SALFA1=DSIN(ALFA1)
      CPSIM=DCOS(PSI1)*SALFA1
      CALFA1=DCOS(ALFA1)
      SIG1=DATAN2(-CALFA1,TPSI1)
      XLB1=DATAN2(-CALFA1,DSIN(PSI1)*SALFA1)
      SPSIM=DSIN(DARCOS(CPSIM))
      XK=FUNK(SPSIM)
      TWSIG1=2*SIG1
      TWS1P=TWSIG1+XK*(DSIN(TWSIG1)-XK*(DSIN(2*TWSIG1)/8.))
      DSP=(1.-XK)*DKM/BRAVO
      TWSP=TWS1P+DSP
      DSIG=DSP+XK*(XK*5*DSIN(2*DSP)*DCOS(2*TWSP)/8.-DCOS(TWSP)*DSIN(DSP)
     1)
      SIG2=SIG1+DSIG
      SIG=(2*SIG1+DSIG)/2.
      XLB2=DATAN2(DSIN(SIG2),DCOS(SIG2)*CPSIM)
      DLB=XLB2-XLB1
      SPSI2=DCOS(SIG2)*SPSIM
      PSI2=DARSIN(SPSI2)
      PHI2=DATAN(DTAN(PSI2)/R1ME2)
      XLN2=XLN1+DLB-CPSIM*(DSIG*(1.-R1ME2)-EPS2*XK*(DSIN(2*DSIG)*DCOS(4*
     1PSI2)*XK/(4*DSIN(DSIG)*DCOS(2*SIG))-DSIG*(1.+XK/2.))*CPSIM/4.)
```

```
      FLAT=PHI2/DETRA
      FLON=XLN2/DETRA
C        TO GET AZ2:
      GO TO 30
C        INVERSE SOLUTION.
  200 XLN2=FLON*DETRA
      DXLN=XLN2-XLN1
      IF(DXLN.GT.PI) DXLN=TUPI-DXLN
      DXLN=DSIGN(DMAX1(DABS(DXLN),DBLE(2.E-4)),DXLN)
      HDLB1=DXLN/2.
      PHI2=FLAT*DETRA
      TPSI2=DTAN(PHI2)*R1ME2
      PSI2=DATAN(TPSI2)
   10 TEMP=DATAN2((TPSI1-TPSI2)/DTAN(HDLB1),TPSI1+TPSI2)
      XLB1=TEMP-HDLB1
      XLB2=TEMP+HDLB1
      CLB1=DCOS(XLB1)
      PSIM=DATAN(TPSI1/CLB1)
      CPSIM=DCOS(PSIM)
      SIG1=DATAN2(CPSIM*DSIN(XLB1),CLB1)
      SIG2=DATAN2(CPSIM*DSIN(XLB2),DCOS(XLB2))
      DSIG=SIG2-SIG1
      SIG=(SIG2+SIG1)/2.
      XK=FUNK(DSIN(PSIM))
      TERM=XK*DCOS(2*SIG)*DSIN(DSIG)
      HDLB2=(DXLN+EPS2*CPSIM*(DSIG*(2./(R1ME2+1.)-XK/2.)-TERM/2.)/2.)/2.
      IF(DABS(HDLB2-HDLB1).LT.1.E-7) GO TO 20
      HDLB1=HDLB2
      GO TO 10
   20 DKM=BRAVO*(DSIG+TERM-(XK**2)*DSIN(2*DSIG)*DCOS(4*SIG)/8.)/(1.-XK)
      ALFA1=DATAN2(ONE,-DTAN(XLB1)*DSIN(PSI1))
      AZ1=ALFA1
      IF(DKM.LT.0) AZ1=AZ1+PI
      IF(LDEG) AZ1=AZ1/DETRA
   30 IF(DABS(DABS(XLB2)-HAPI).GE.1.E-7) GO TO 35
      ALFA2=ALFA1
      GO TO 36
   35 ALFA2=DATAN2(ONE,DTAN(XLB2)*SPSI2)
   36 AZ2=ALFA2
      IF(LDEG) AZ2=AZ2/DETRA
      DKM=DABS(DKM)
      RETURN
      END
C/*
//LINK.USDD DD UNIT=3330,VOL=SER=DISK01,DSN=S2648.ECTLMOD,
// DISP=SHR,LABEL=(,,,IN)
//LINK.SYSIN DD *
 INCLUDE USDD(GEDSCT)
```

```
 ENTFY MAIN
C/*
//GC.FTC1FC01  CC CSN=S2648.GRID2,UNIT=3330,VCL=SER=DISK01,
//   CISP=SHR,LABEL=(,,,IN)
//GC.SYSIN  CC  *
  -10     5
    C.     4.     C.     0.
     1  151    15    23
C/*
```

GEODST

```
C
C
C
C
C                             GEODST
C
C
C*****************************************************************
C
C       SUBROUTINE GEODST IS USED IN PROGRAMS GENBOT AND GRDCHK
C
C
      SUBROUTINE GEODST(SLAT,SLON,FLAT,FLON,AZ1,AZ2,DKM,MODE)
C
C
C        SOURCE/DATE:     L. R. ERAZO, JAN 1980
C        PURPOSE:  SOLVES FORWARD AND INVERSE GEODESY NAVIGATING
C        TRIANGLE, CALCULATING GREAT CIRCLE DISTANCES, COORDINATES AND
C        AZIMUTHS TO A GREATER ACCURACY THAN ATTAINED FROM SPHERICAL
C        TRIGONOMETRY.  USES GEODETIC CONSTANTS FROM WGS 72.
C        REFERENCE:      RAND CORPORATION PUBLICATION P-4970
C        USAGE:
C        SLAT        LATITUDE OF FIRST STATION
C        SLON        LONGITUDE OF FIRST STATION
C        FLAT        LATITUDE OF SECOND STATION
C        FLON        LONGITUDE OF SECOND STATION
C        AZ1         AZIMUTH FROM STATION 1 TO STATION 2,
C                    MEASURED FROM TRUE NORTH
C        AZ2         BEARING, LT 180 DEGREES, MEASURED CLOCKWISE
C                    FROM HEADING AT STATION 2 TO LOCAL MERIDIAN
C        DKM         DISTANCE IN KILOMETERS
C        MODE        IF LT 0, SOLVES INVERSE PROBLEM (REQUIRES
C                    SLAT, SLON, FLAT, FLON) ;
C                    IF GE 0, SOLVES FORWARD PROBLEM (REQUIRES
C                    SLAT, SLON, AZ1, DKM)
C                    IF IABS(MODE) = 1, AZIMUTHS ARE MEASURED IN RADIANS,
C                    OTHERWISE IN DEGREES.
C
C                    ALL GEOGRAPHIC COORDINATES ARE MEASURED IN DEGREES
C                    AND DECIMALS OF DEGREES
C                    ALL FLOATING POINT NUMBERS ARE DOUBLE PRECISION
C                    (REAL*8)
      IMPLICIT REAL*8(A-H),REAL*8(O-Z)
      LOGICAL LDEG
      DATA EPS/.081819356/,BRAVO/6356.75023/
      FUNK(SPSI)=DTAN(DATAN(EPS*SPSI/R1ME2)/2.)**2
      ONE=1.
      HAPI=DATAN(ONE)*2
```

```
      PI=HAPI*2
      TUPI=2*PI
      DETRA=PI/180.
      EPS2=EPS**2
      R1ME2=DSQRT(ONE-EPS2)
      IF(SLAT.EQ.0.) SLAT=1.E-5
      PHI1=SLAT*DETRA
      XLN1=SLON*DETRA
      IF(DABS(SLAT).LT.90.) GO TO 1
      TPSI1=DSIGN(DBLE(5.E9),SLAT)
      GO TO 2
    1 TPSI1=DTAN(PHI1)*R1ME2
    2 PSI1=DATAN(TPSI1)
      LDEG=IABS(MODE).NE.1
      DKM=DABS(DKM)
      IF(MODE) 200,100,100
C        FORWARD SOLUTION.
  100 ALFA1=AZ1
      IF(LDEG) ALFA1=AZ1*DETRA
      IF(ALFA1.EQ.TUPI) ALFA1=0.
      IF(ALFA1.LT.0.) ALFA1=ALFA1+TUPI
      IF(ALFA1.LE.PI) GO TO 101
      DKM=-DKM
      ALFA1=ALFA1-PI
  101 SALFA1=DSIN(ALFA1)
      CPSIM=DCOS(PSI1)*SALFA1
      CALFA1=DCOS(ALFA1)
      SIG1=DATAN2(-CALFA1,TPSI1)
      XLB1=DATAN2(-CALFA1,DSIN(PSI1)*SALFA1)
      SPSIM=DSIN(DARCOS(CPSIM))
      XK=FUNK(SPSIM)
      TWSIG1=2*SIG1
      TWS1P=TWSIG1+XK*(DSIN(TWSIG1)-XK*(DSIN(2*TWSIG1)/8.))
      DSP=(1.-XK)*DKM/BRAVO
      TWSP=TWS1P+DSP
      DSIG=DSP+XK*(XK*5*DSIN(2*DSP)*DCOS(2*TWSP)/8.-DCOS(TWSP)*DSIN(DSP
     1))
      SIG2=SIG1+DSIG
      SIG=(2*SIG1+DSIG)/2.
      XLB2=DATAN2(DSIN(SIG2),DCOS(SIG2)*CPSIM)
      DLB=XLB2-XLB1
      SPSI2=DCOS(SIG2)*SPSIM
      PSI2=DARSIN(SPSI2)
      PHI2=DATAN(DTAN(PSI2)/R1ME2)
      XLN2=XLN1+DLB-CPSIM*(DSIG*(1.-R1ME2)-EPS2*XK*(DSIN(2*DSIG)*DCOS(4*
     1PSI2)*XK/(4*DSIN(DSIG)*DCOS(2*SIG))-DSIG*(1.+XK/2.))*CPSIM/4.)
      FLAT=PHI2/DETRA
      FLON=XLN2/DETRA
```

```
C        TO GET AZ2:
      GO TO 30
C        INVERSE SOLUTION.
  200 XLN2=FLON*DETRA
      DXLN=XLN2-XLN1
      IF(DXLN.GT.PI) DXLN=TLPI-DXLN
      DXLN=DSIGN(DMAX1(DABS(DXLN),DBLE(2.E-4)),DXLN)
      HDLB1=DXLN/2.
      PHI2=FLAT*DETRA
      TPSI2=DTAN(PHI2)*R1ME2
      PSI2=DATAN(TPSI2)
   10 TEMP=DATAN2((TPSI1-TPSI2)/DTAN(HDLB1),TPSI1+TPSI2)
      XLB1=TEMP-HDLB1
      XLB2=TEMP+HDLB1
      CLB1=DCOS(XLB1)
      PSIM=DATAN(TPSI1/CLB1)
      CPSIM=DCOS(PSIM)
      SIG1=DATAN2(CPSIM*DSIN(XLB1),CLB1)
      SIG2=DATAN2(CPSIM*DSIN(XLB2),DCOS(XLB2))
      DSIG=SIG2-SIG1
      SIG=(SIG2+SIG1)/2.
      XK=FLNK(DSIN(PSIM))
      TERM=XK*DCOS(2*SIG)*DSIN(DSIG)
      HDLB2=(DXLN+EPS2*CPSIM*(DSIG*(2./(R1ME2+1.)-XK/2.)-TERM/2.)/2.)/2.
      IF(DABS(HDLB2-HDLB1).LT.1.E-7) GO TO 20
      HDLB1=HDLB2
      GO TO 10
   20 DKM=BRAVG*(DSIG+TERM-(XK**2)*DSIN(2*DSIG)*DCOS(4*SIG)/8.)/(1.-XK)
      ALFA1=DATAN2(ONE,-DTAN(XLB1)*DSIN(PSI1))
      AZ1=ALFA1
      IF(DKM.LT.0) AZ1=AZ1+PI
      IF(LDEG) AZ1=AZ1/DETRA
   30 IF(DABS(DABS(XLB2)-HAPI).GE.1.E-7) GO TO 35
      ALFA2=ALFA1
      GO TO 36
   35 ALFA2=DATAN2(ONE,DTAN(XLB2)*SPSI2)
   36 AZ2=ALFA2
      IF(LDEG) AZ2=AZ2/DETRA
      DKM=DABS(DKM)
      RETURN
C     DEBUG INIT(HDLB1,CLB1,XK)
      END
```

```
C
C
C
C                               APPENDIX M
C
C
C
C
C************************************************************************
C
C
C
C
C                               ECTRACE
C
C
C
C
C************************************************************************
C
C ECTRACE  JOB  (    ,0354,IX83),'ERAZO  SMC1704',TIME=5
//   EXEC  FORTCLGW,REGION.GO=250K
//FORT.SYSIN  DD  *
C        UNIVERSAL MAIN PROGRAM                                          ECS00010
C        ESTABLISHES SIZE OF DEPTH MATRIX IN LIEU OF OBJECT
C        TIME DIMENSIONING.                                              ECS00030
      COMMON /DIMS/ KMX,KMY
      DIMENSION ZB(151,151)                                              ECS00050
      KMX=151                                                            ECS00060
      KMY=151                                                            ECS00070
      CALL TRACER(ZB)                                                    ECS00090
      STOP                                                               ECS00100
      END                                                                ECS00110
C/*
//LINK.USDD  DD  UNIT=3330,VOL=SER=DISK01,DSN=S2648.ECTLMOD,
//     DISP=SHR,LABEL=(,,,IN)
//LINK.SYSIN  DD  *
 INCLUDE USDD(TRACER,IDPROF,NUPROF,CHNLIM)
 INCLUDE USDD(CONTAC,BOANG,ANGPRT,IDTSUB,DEEP)
 INCLUDE USDD(EGNPLT,PDTPLT,RNGPLT,T2DPLT,ENDPLT)
 INCLUDE USDD(GRDSCT,SSPPLT,FBLOSS)
 ENTRY MAIN
C/*
//*REMOVE OR NULLIFY THE FOLLOWING CARD TO PRODUCE PUNCHED SUMMARY
//GO.FT07F001  DD  DUMMY
//GO.FT01F001  DD DSN=S2648.GRID2,UNIT=3330,VOL=SER=DISK01,
//   DISP=SHR,LABEL=(,,,IN)
//GO.SYSIN  DD  *
C
```

```
C
C         REFER TO CHAPTER V 'USER INSTRUCTIONS' FOR A
C         DESCRIPTION OF THE FOLLOWING 5 DATA INPUT CARDS.
C
C
           TRIAL 84  *** GENBCT GENERATED GRID2 ***
   25.     125.     800.     125.      75.   5000.    150.    150.
    0.       0.       0.       5.     -10.    125.
  015.     125.     -10.      10.     200.      1.
    9     3   21    10     6     3
C
C     FIRST WATER MASS SOUND SPEED PROFILE
NWMR  9
    0.   1461.      50.   1461.      60.   1447.      70.   1445.      90.   1447.
  500.   1454.3  1050.    1461.   1500.    1470.    4000.   1511.7
C     BOUNDARY BETWEEN FIRST AND SECOND WATER MASS
  FENTX    FENTY    FEXTX    FEXTY
    0.     140.      10.     150.
C
C     SECOND WATER MASS SOUND SPEED PROFILE
SEMR  9
    0.   1490.      50.   1482.     200.   1474.5    375.   1474.     750.   1466.
  975.   1465.   1250.    1467.   1500.    1470.    4000.   1511.7
C     BOUNDARY BETWEEN SECOND AND THIRD WATER MASS
  FENTX    FENTY    FEXTX    FEXTY
    0.       0.     150.      20.
C
C     THIRD WATER MASS SOUND SPEED PROFILE
ISCV  2
    0.   1470.   5000.   1470.
C/*
C
C         LISTING FOR GRDSCT MAY BE FOUND IN G3CP APPENDIX
C
```

```
C
C
C
C
C                                   TRACER
C
C
C
C
C*****************************************************************************
C
      SUBROUTINE TRACER(ZE)
C        SOURCE:   L. R. ERAZO
C        PURPOSE:  READS ALL USER INPUT DATA, PERFORMS ACTUAL RAY
C              TRACING AND OUTPUTS TRACE HISTORY.   INVOKES PLOT ROUTINES.
      IMPLICIT REAL*8(C-D),REAL*8(G),REAL*8(P-W)
      IMPLICIT LOGICAL*4(L)
      COMMON /DIMS/ KMX,KMY
      COMMON /PRINTS/ FBOT(30,50),FANGLE(30,50)
      COMMON /PARAM/ C(50,3),GRADV(50,3),PIOT,ZC(50,3)
      COMMON /FLCTII/ TTL,CNX,CNY,CNLAT,CNLON
      COMMON /PROFIL/ OMAS(3),LNOT(2),LEW(2),LNS(2),FENTY(2),FENTX(2),
     1    FN(2)
      COMMON /SUBS/ CX,CY,ILT,IRT,IVR,JBT,JTP,JVR,XMIN,YMIN
      COMMON /CHANEL/ CO,CRE,CRN,VV,DEP1,CHS,CHD,
     1    ZEND,LCAL(3),NVAL(3),KUP,KLW,ISSP,KMAX
      DIMENSION TTL(12),ZE(KMX,KMY),OTIT(20)
      DIMENSION XFUN(103),YFUN(103)
      DIMENSION CHD(100),CEL(100),NRB(100),OBB(100),OTT(100),ORM(100)
      DIMENSION FEXTX(2), FEXTY(2), FXGAP(2), FYGAP(2)
      DATA OSURF/'SURF'/,OBOTT/'BOTT'/,OINIT/'INIT'/,FEND/9999./,
     1     OOVER/'OVER'/,OUNDR/'UNDR'/,OSTOP/'STOP'/,OCHCK/'    '/
       READ 5000,OTIT
 5000  FORMAT(20A4)
       PRINT 6000,OTIT
 6000  FORMAT(//,10X,20A4,//)
       PUNCH 5000,OTIT
C
C        SEE DOCUMENTATION FOR DESCRIPTION OF INPUT PARAMETERS.
      READ 5010,XO,YO,ZO,XREC,YREC,ZREC,FREQ,ELMAX,FDL
      READ 5010,XMIN,YMIN,ZST,ZEND,ARST,AREND
      READ 5010,HDST,HDEND,ELST,ELEND,HORLIM,SMNGL
      READ 5020,NHD,NEL,NSTOP,INCPLT,NLEN,NSSF
 5005 FORMAT(A4,I3)
 5010 FORMAT(10F8.2)
 5020 FORMAT(16I5)
      HDWID=HDEND-HDST
      ELWID=ELEND-ELST
      IF(HDWID.LT.0.) HDWID=HDWID+360.
```

```
      HDDV=0.
      ELDV=0.
      NHD=MIN0(NHD,30)
      NEL=MIN0(NEL,100/NHD)
      NSTOP=MIN0(NSTOP,50)
      IF(NHD.GT.1) HDDV=HDWID/FLOAT(NHD-1)
      IF(NEL.GT.1) ELDV=ELWID/FLOAT(NEL-1)
C
C        MEAN HEADING OF RAYFANS
      DIRA=HDST+HDDV*FLOAT(NHD-1)/2.
      LOSS=ELMAX.LT.1000
      IF(SMNGL.EQ.0.) SMNGL=2.
C
C        FDL DEFAULTS TO TEN WAVELENGTHS.
      WVLN=1.5/FREQ
      IF(FDL.EQ.0.) FDL=10000*WVLN
C
C        INPUT WATER MASS DATA
      MAXV=1
      DO 20 J=1,3
         IF(J.LE.NSSP) GO TO 10
         NVAL(J)=0
         GO TO 20
   10    LCAL(J)=.FALSE.
         READ 5005, OMAS(J), KMAX
         NVAL(J)=KMAX
         MAXV=MAX0(MAXV,KMAX)
         READ 5010, (ZC(I,J),C(I,J),I=1,KMAX)
         IF(J.EQ.NSSP) GO TO 20
         READ 5010, FENTX(J),FENTY(J),FEXTX(J),FEXTY(J)
         FYGAP(J)=FEXTY(J)-FENTY(J)
         FXGAP(J)=FEXTX(J)-FENTX(J)
         LEW(J)=FYGAP(J).EQ.0.
         LNS(J)= FXGAP(J).EQ.0.
         LNCT(J)=LEW(J).AND.LNS(J)
         IF(LNCT(J)) GO TO 20
         IF(LEW(J).OR.LNS(J)) GO TO 20
         FM(J)=FYGAP(J)/FXGAP(J)
   20    CONTINUE
C
C        CALCULATE GRADIENTS AND OUTPUT WATER MASS DATA.
      DO 40 J=1,NSSP
      KMAX=NVAL(J)
      KE=KMAX-1
      DO 30 I=1,KE
      K=I+1
   30 GRADV(I,J)=(C(K,J)-C(I,J))/(ZC(K,J)-ZC(I,J))
   40 GRADV(KMAX,J)=.01668
```

```
      PRINT 6040
      DO 50 I=1,MAXV
      IF(I.GT.NVAL(1)) GO TO 41
      PRINT 6041, ZC(I,1), C(I,1), GRADV(I,1)
   41 IF(I.GT.NVAL(2)) GO TO 42
      PRINT 6042, ZC(I,2), C(I,2), GRADV(I,2)
   42 IF(I.GT.NVAL(3)) GO TO 50
      PRINT 6043, ZC(I,3), C(I,3), GRADV(I,3)
   50 PRINT 6050
 6050 FORMAT(' ')
 6041 FORMAT('+',12X,F8.2,5X,F8.2,3X,F8.5)
 6042 FORMAT('+',51X,F8.2,5X,F8.2,3X,F8.5)
 6043 FORMAT('+',90X,F8.2,5X,F8.2,3X,F8.5)
      PRINT 6070, (CMAS(I), I=1,NSSP)
      NFRO=NSSP-1
      PRINT 6080, (FENTX(I),FENTY(I),FEXTX(I),FEXTY(I),I=1,NFRO)
 6040 FORMAT(' SOUND SPEED PROFILES'//6X,3(13X,'ZC',7X,' C(ZC)',
     1   7X,'GRAD')//)
 6070 FORMAT(3(22X,A4,13X))
 6080 FORMAT('-FRONT/BOUNDARY INTERSECTIONS'//T12,2(27X,'FRONT  ')/T22,
     1   2(8X,'ENTRY',14X,'EXIT    ')/T23,4(2X,'(',F6.2,',',F6.2,')'))
      LNCR=(XREC.EQ.XO.AND.YREC.EQ.YO)
      ZREC=ZREC/1000.
      ZO=ZO/1000.
C
C        INPUT DEPTH VALUE GRID FROM FILE.
      CALL GRDSCT(XMIN,YMIN,XMAX,YMAX,IMAX,JMAX,CHN,ZB)
      PUNCH 7010,TTL
 7010 FORMAT(6A8/6A8)
      XO=AMAX1(XO,XMIN)
      XO=AMIN1(XO,XMAX)
      YO=AMAX1(YO,YMIN)
      YO=AMIN1(YO,YMAX)
C
C        PULL SOURCE OUT OF MUD.
      CRE=XO
      CRN=YO
      WHO=DEEP(CRE,CRN,CHN,ZB,IER)
      IF(ZO.GT.WHO) ZO=WHO-.001
      PIOT = DATAN(DBLE(1.))
      PI = 4 * PIOT
      TUPI = 2 * PI
      DETRA = PI/180.
C
C        MINIMUM DEPTH MUST BE AT LEAST TEN WAVELENGTHS.
      HSTOP=10.*WVLN
C
C        ATTEMPT TO PUT A MISPLACED SOURCE INTO GRID, UNDER WATER.
```

```
      DIR=DIRA*DETRA                                                      TFA01480
      IF(DIR.GT.PI) DIR=DIR-TUPI                                          TFA01490
      DIRX=DHN*DSIN(DIR)                                                  TFA01500
      DIRY=DHN*DCOS(DIR)                                                  TFA01510
   80    IF(WHO.GT.HSTOP.AND.Z0.GT.0.) GO TO 90                           TFA01520
         CRE=CRE+DIRX                                                     TFA01530
         CRN=CRN+DIRY                                                     TFA01540
         WHO=DEEP(CRE,CRN,DHN,ZB,IER)                                     TFA01550
         IF(IER.NE.0) GO TO 85                                            TFA01560
         Z0=WHO-.001                                                      TFA01570
         GO TO 80                                                         TFA01580
   85 PRINT 6160                                                          TFA01590
 6160 FORMAT(/T25,'***ERROR*** PROPAGATION IMPOSSIBLE FROM GIVEN SOURCE  TFA01600
     1POSITION AND DEPTH VALUES.'/T25,'RELOCATION ATTEMPTS FAILED.')      TFA01610
      RETURN                                                              TFA01620
   90 X0=CRE                                                              TFA01630
      Y0=CRN                                                              TFA01640
      IF(LNOF) ZREC=Z0                                                    TFA01650
C                                                                         TFA01660
C        ECHO PRINT INPUT PARAMETERS.                                     TFA01670
      PRINT 6090                                                          TFA01680
      PRINT 6100, FREQ                                                    TFA01690
      PRINT 6110, X0,Y0,Z0                                                TFA01700
      PRINT 6120,NEL,ELST,ELEND,ELDV,NHD,HDST,HDEND,HDDV                  TFA01710
      PRINT 6130                                                          TFA01720
      PRINT 6110, XREC, YREC,ZREC                                         TFA01730
      PRINT 6140, KMX, KMY, DHN                                           TFA01740
      PRINT 6150, ELMAX, NSTOP, PDL, HORLIM, SMNGL                        TFA01750
 6090 FORMAT('OTHE FOLLOWING ADDITIONAL PARAMETERS WERE INPUT:')          TFA01760
 6100 FORMAT(/T6,'SOURCE DATA:',T25,'FREQUENCY ',F7.2,' HZ,'/)            TFA01770
 6110 FORMAT( '+',T25,'GRID COORDINATES (',F8.2,',',F8.2,'), DEPTH ',     TFA01780
     1   F7.3,' KM.')                                                     TFA01790
 6120 FORMAT( T25,I2,' RAYFANS WILL BE TRACED, FROM ',F5.1,' TO ',F5.1,  TFA01800
     1   ' DEGREES ELEVATION, INCREMENTING EVERY ',F5.1,' DEGREES.'/     TFA01810
     2   T25,I2,' RAYS MAKE A RAYFAN FROM ',F5.1,' TO ',F5.1,' DEGREES TTFA01820
     3RUE HEADING, INCREMENTING EVERY ',F5.1,' DEGREES.')                 TFA01830
 6130 FORMAT(/T6,'RECEIVER DATA:')                                        TFA01840
 6140 FORMAT(/T6,'GRID DATA:',T25,'MAXIMUM ARRAY SIZE (',I3,',',I3,')'/ TFA01850
     1   T25,'DEPTH VALUES RESOLVED BY ',F4.1,' KM SPACING.')             TFA01860
 6150 FORMAT(/T6,'TRACE PARAMETERS:',T25,'PROPAGATION WILL CONTINUE UNTITFA01870
     1L A BOTTOM LOSS OF ',F5.1,' DB OR ',I4,' BOTTOM CONTACTS EXCEEDED TFA01880
     2FOR EACH RAY.'/T25,'PATH INCREMENT IS ',F6.1,' M.'/T25,'CHANNEL RATFA01890
     3NGE LIMIT ',F5.1,' KM. SMALL ANGLE=',F5.2,' DEGREES.')              TFA01900
      PRINT 6180                                                          TFA01910
      LPLOT=NLEN.GE.4                                                     TFA01920
      IF(LPLOT) GO TO 100                                                 TFA01930
      PRINT 6190                                                          TFA01940
      GO TO 110                                                           TFA01950
```

```
C
C        ESTABLISH PLOT PARAMETERS.
  10C NLEN=MINO(NLEN,19)
      FACT=NLEN/15.
      AXY=AMAX1(15.,AMAX1(XMAX-XMIN,YMAX-YMIN))
      AR15=AXY/15.
      FDV=AINT(AR15)
      AFACT=FDV/AR15
      FACT=FACT*AFACT
      PUNCH 7020,XMIN,YMIN,FDV,CNX,CNY,CNLAT,CNLCN,FACT
 7020 FORMAT(10F8.3)
      ZDV=AINT((ZEND-ZST)/3.5)
      IF(AMOD(ZEND-ZST,3.5).NE.0.) ZDV=ZDV+1.
      IF(ZST.EQ.0.) ZST=-.001
      ZEND=ZST+3.5*ZDV
      ARDV=AINT((AREND-ARST)/19.)
      IF(AMOD(AREND-ARST,19.).NE.0.) ARDV=ARDV+1.
      AREND=ARST+19.*ARDV
      PRINT 6170,XMIN,YMIN,ZST,ARST,FDV,FDV,ZDV,ARDV,XMAX,YMAX,ZEND,
     1   AREND,NLEN,FACT
      INCPLT=MAXO(1,INCPLT)
      PRINT 6200, INCPLT
 6170 FORMAT('+',T18,'SELECTED.'/T6,'PLOT PARAMETERS:',T37,'X',T47,'Y',
     1   T57,'Z',T67,'R'/T13,'AXES:',
     2   T25,'START',T32,4(F8.2,2X)/T25,'DELTA',T32,4(F8.2,2X)/T25,
     3   'END',T32,4(F8.2,2X)//T13,'SCALING:',T25,'LENGTH',I3,' IN., FAC
     4TOR=',F6.4)
 6180 FORMAT(/T6,'PLOT OPTION ')
 6190 FORMAT('+',T18,'REFUSED.')
 6200 FORMAT(/T6,'EACH SEGMENT REPRESENTS',I3,' POSITION CALCULATIONS.')
C
C        INITIAL RECEIVER DISTANCE.
      XDIS=XREC-XO
      YDIS=YREC-YO
      STR=ATAN2(XDIS,YDIS)
      DBR=STR-DIR
      IF(DBR.LT.-PI) DBR=DBR+TWPI
      DIST=SQRT(XDIS**2+YDIS**2)
      AREC=DIST*COS(DBR)
C
C        PLOT INITIALIZATION ROUTINES
      CALL BGNPLT(OTIT,XMIN,YMIN,FDV,ARST,ARDV,ZST,ZDV,ZEND,
     1   XREC,YREC,AREC,ZREC,FACT)
      CALL BOTPLT(DIRX,DIRY,DHN,WHO,XO,YO,ZEND,XMIN,XMAX,YMIN,YMAX,ZST,
     1   ZDV,ARST,ARDV,ZB)
      CALL SSPPLT(NVAL,NSSP,CMAS)
      KRAY=0
      FCL=FCL/1000.
```

```
      IEL=0                                                             TFA02440
      SMNGL=SMNGL*DETRA                                                 TFA02450
C                                                                       TFA02460
C        BEGIN NEXT RAYFAN.                                             TFA02470
  110 IEL=IEL+1                                                         TFA02480
      IMK=MOD(IEL-1,13)                                                 TFA02490
      NBNCF=0                                                           TFA02500
      IHD=0                                                             TFA02510
      DO 120 I=1,NHD                                                    TFA02520
      DO 120 J=1,50                                                     TFA02530
      FBCT(I,J)=0.0                                                     TFA02540
      FANGLE(I,J)=0.0                                                   TFA02550
  120 CONTINUE                                                          TFA02560
      ELEVO=ELST+(IEL-1)*ELDV                                           TFA02570
      THO=ELEVO*DETRA                                                   TFA02580
C                                                                       TFA02590
C        BEGIN NEXT RAY IN RAYFAN. NBNC COUNTS BOTTOM BOUNCES,          TFA02600
C        NREV COUNTS ALL REVERSALS.                                     TFA02610
  130 KRAY=KRAY+1                                                       TFA02620
      ISSP=0                                                            TFA02630
      NBNC=0                                                            TFA02640
      NCTFL=0                                                           TFA02650
      NREV=-1                                                           TFA02660
      BTMLSS=0.0                                                        TFA02670
      LREV=.TRUE.                                                       TFA02680
      LHORZ=.FALSE.                                                     TFA02690
      LLR=.FALSE.                                                       TFA02700
      LBB=.TRUE.                                                        TFA02710
      NPTS=1                                                            TFA02720
      NSEG=-1                                                           TFA02730
      FRANGE=0.                                                         TFA02740
      DR=0.                                                             TFA02750
      ELAPS=0.                                                          TFA02760
      WH1=WHO                                                           TFA02770
      DEP1=ZO                                                           TFA02780
      DTYPE=DINIT                                                       TFA02790
      IHD=IHD+1                                                         TFA02800
C                                                                       TFA02810
C         INITIAL HEADING.                                              TFA02820
      AHDGO=HDST+(IHD-1)*HDDV                                           TFA02830
      IF(AHDGO.LE.0.)    AHDGO=AHDGO+360.                               TFA02840
      IF(AHDGO.GT.360.)  AHDGO=AHDGO-360.                               TFA02850
      AHDG=AHDGO                                                        TFA02860
      CHD(KRAY)=AHDG                                                    TFA02870
      CEL(KRAY)=ELEVO                                                   TFA02880
      TH1=THO                                                           TFA02890
      PHI=AHDG*DETRA                                                    TFA02900
      IF(PHI.GT.PI) PHI=PHI-TUPI                                        TFA02910
```

```
C                                                                        TFA02920
C         HORIZONTAL DIRECTION VECTOR.                                  TFA02930
      UVX=DSIN(PHI)                                                      TFA02940
      UVY=DCOS(PHI)                                                      TFA02950
C                                                                        TFA02960
C         HORIZONTAL TRACERS.                                            TFA02970
      CRE=XO                                                             TFA02980
      CRN=YO                                                             TFA02990
      CRMIN=1000.                                                        TFA03000
      PRINT 6220,KRAY,IHD,IEL                                            TFA03010
      PRINT 6230                                                         TFA03020
      PRINT 6240                                                         TFA03030
      PRINT 6250                                                         TFA03040
 6220 FORMAT('1RAY ',I3,' (',I3,' OF RAYFAN ',I3,') TRACE HISTORY'/     TFA03050
     1'OREV',T15,'X',T24,'Y     DEPTH   GRID      ELEV   BRNG     DIST', TFA03060
     2T77,'WTR       DEPTH LIMITS     C')                                TFA03070
 6230 FORMAT('+',T108,'GRAZ',2(4X,'BTM'))                                TFA03080
 6240 FORMAT(' NO. TYPE    (KM)     ',2('  (KM)   '),                    TFA03090
     1'HEADING  ANGLE  TO REC  FM REC   TIME   MASS LYR  MIN    MAX   (KTFA03100
     2M/S)')                                                             TFA03110
 6250 FORMAT('+',T107,2(' ANGLE  '),'LOSS LPS'/1X,129('-'))              TFA03120
C                                                                        TFA03130
C         HORIZONTAL PLOT POINT.                                         TFA03140
  150 XPLN(NPTS)=CRE                                                     TFA03150
      YPLN(NPTS)=CRN                                                     TFA03160
C                                                                        TFA03170
C         DEPTH DIRECTION VECTOR.                                        TFA03180
      CT1=DCOS(TH1)                                                      TFA03190
      ST1=DSIN(TH1)                                                      TFA03200
C                                                                        TFA03210
C         TEST BOTTOM DEPTH AND ITERATE NEXT RAY SEGMENT.                TFA03220
  160 IF(WH1.LT.HSTOP.OR.WH1.GE.10.) GO TO 530                           TFA03230
  164 CALL IDPROF(LNUP)                                                  TFA03240
      IF(LNUP) GO TO 165                                                 TFA03250
      IF(LLR) GO TO 170                                                  TFA03260
      IF(.NOT.LBB) GO TO 180                                             TFA03270
C                                                                        TFA03280
C         RAY TAKES ON NEW PARAMETERS FROM NEW WATER MASS                TFA03290
C         OR A BOTTOM BOUNCE.                                            TFA03300
  165 C1=CC                                                              TFA03310
      HDIS=0.                                                            TFA03320
C                                                                        TFA03330
C         VERTEX VELOCITY (RAY INVARIANT).                               TFA03340
      VV=C1/CT1                                                          TFA03350
      CALL CHNLIN                                                        TFA03360
  170 G=GRADV(KUP,ISSP)                                                  TFA03370
      DL=FDL                                                             TFA03380
      IF(KLW.LE.KMAX)DL=AMIN1(FDL,(ZC(KLW,ISSP)-ZC(KUP,ISSP)))           TFA03390
```

```
      LRFR=(CT1.NE.0.).AND.(G.NE.0.)
      IF(.NOT.LRFR) GO TO 180
C
C        TRAVEL TIME INTEGRATION LIMITS.
      W2P=C1/G
      W2=W2P
C
C        RADIUS OF CURVATURE.
      RHO=-VV/G
      DTH=DL/RHO
C
C        MAX THETA INCREMENT IS .001
      DTH=DSIGN(DMIN1(DABS(DTH),DBLE(.001)),DTH)
      CDTH=DABS(DTH)
      LLR=.FALSE.
C
C        DISTANCE CALCULATIONS.
  180 XDIS=CRE-XREC
      YDIS=CRN-YREC
      ZDIS=DEP1-ZREC
      RSQ=XDIS**2+YDIS**2
      IF(RSQ.NE.0.) GO TO 181
      BTR=0.
      GO TO 182
  181 BTR=ATAN2(XDIS,YDIS)/DETRA+180.
      IF(BTR.GT.360.) BTR=BTR-360.
      IF(BTR.LE.0.) BTR=BTR+360.
  182 DIST=DSQRT(RSQ+(ZDIS**2))
      IF(LNCR.OR.DIST.GE.CRMIN) GO TO 183
C
C        MINIMUM DISTANCE WAS DETECTED.
      CTHD=AHDG
      CTIME=ELAPS
      CRMIN=DIST
  183 IF(LREV) GO TO 190
      IF(MOD(NSEG,INCPLT).EQ.0) GO TO 195
      GO TO 200
C
C        REVERSAL POINT.
  190 NREV=NREV+1
      ELEV=TH1/DETRA
      PRINT 5040,NREV,OTYPE,CRE,CRN,DEP1,AHDG,ELEV,BTR,DIST,ELAPS,
     1   GMAS(ISSP),KUP,CHS,CHC,C1
 5040 FORMAT(1X,I3,1X,A4,2F9.3,F7.3,5F8.3,1X,A4,I4,2(2X,F5.3),F8.5)
      IF(OTYPE.NE.OBOTT) GO TO 185
      BNGL=180.-BANGLE
      PRINT 5060,FRAZ,BNGL,BTMLSS,NLPS
 5060 FORMAT('+',T106,2F8.3,F6.1,I3)
```

```
  185 IF(NBNC.GE.NSTOP.OR.ETMLSS.GE.BLMAX) GO TO 510                        TRA03880
      IF(LBB) HDIS=0.                                                       TRA03890
      IF(HDIS.GE.HORLIM) GO TO 550                                          TRA03900
      LREV=.FALSE.                                                          TRA03910
      LHORZ=.FALSE.                                                         TRA03920
      IF(LRFR) W2=W2P                                                       TRA03930
  195 NCTPL=NCTPL+1                                                         TRA03940
      IF(.NOT.LPLOT) GO TO 200                                              TRA03950
C                                                                           TRA03960
C        PLOT DEFINITION POINT.                                             TRA03970
      IF(LNCR) GO TO 191                                                    TRA03980
      XDIS=CRE-XO                                                           TRA03990
      YDIS=CRN-YO                                                           TRA04000
      RSQ=XDIS**2+YDIS**2                                                   TRA04010
  191 IF(RSQ.NE.0.) GO TO 192                                               TRA04020
      STB=DIR                                                               TRA04030
      GO TO 193                                                             TRA04040
  192 STB=ATAN2(XDIS,YDIS)                                                  TRA04050
C                                                                           TRA04060
C        PROJECT SEGMENT ONTO VERTICAL PLOT PLANE.                          TRA04070
  193 DIF=STB-DIR                                                           TRA04080
      IF(DIF.LT.-PI) DIF=DIF+TUPI                                           TRA04090
      COSDIF=DCOS(DIF)                                                      TRA04100
      RNG=DSQRT(RSQ)                                                        TRA04110
      FDEPTH=(ZEND-DEP1)/ZDV                                                TRA04120
      FRANGE = (RNG*COSDIF-ARST)/ARDV                                       TRA04130
      CALL RNGPLT(FRANGE,FDEPTH,OTYPE,LBB,IMK)                              TRA04140
C                                                                           TRA04150
C        SORRY, ONLY 200 REVERSALS OR 100 BOTTOM BOUNCES PERMITTED.         TRA04160
  200 IF(NREV.EQ.200.OR.NBTS.EQ.101) GO TO 540                              TRA04170
      LBB=.FALSE.                                                           TRA04180
      IF(LRFR) GO TO 225                                                    TRA04190
C                                                                           TRA04200
C        STRAIGHT RAY SEGMENT.                                              TRA04210
      CLF=CL                                                                TRA04220
      TH2=TH1                                                               TRA04230
      CT2=CT1                                                               TRA04240
      ST2=ST1                                                               TRA04250
      DZ=DLP*ST1                                                            TRA04260
  220 DR=DLP*CT1                                                            TRA04270
      GO TO 250                                                             TRA04280
C                                                                           TRA04290
C        REFRACTED RAY SEGMENT.                                             TRA04300
  225 IF(TH1) 226,235,226                                                   TRA04310
  226 IF(DABS(TH1)-DDTH) 230,235,235                                        TRA04320
C        STOP TH2 AT TURNAROUND POINT.                                      TRA04330
  230 TH2=0.                                                                TRA04340
      CT2=1.                                                                TRA04350
```

```
      ST2=0.                                                            TFA04360
      IF(LHORZ) GO TO 245                                               TFA04370
      GO TO 240                                                         TFA04380
C         NEW ELEVATION ANGLE.                                          TFA04390
  235 TH2=TH1+DTH                                                       TFA04400
      CT2=DCOS(TH2)                                                     TFA04410
      ST2=DSIN(TH2)                                                     TFA04420
C         DIFFERENCE IN COSINES.  CHOOSE ALTERNATE CALCULATION          TFA04430
C         TH1 IS SMALL.                                                 TFA04440
  240 DCT=CT1-CT2                                                       TFA04450
      IF(DABS(TH1).LE.SMAGL) DCT=(TH2**2-TH1**2)/2.-(TH2**4-TH1**4)/24. TFA04460
      DZ=RHO*DCT                                                        TFA04470
  245 DR=RHO*(ST2-ST1)                                                  TFA04480
C                                                                       TFA04490
C         INTERIM DEPTH AND HORIZONTAL COMPONENTS OF RAY SEGMENT HEAD.  TFA04500
  250 DX=DR*LVX                                                         TFA04510
      DY=DR*LVY                                                         TFA04520
      DEP2=DEP1+DZ                                                      TFA04530
      IF(LBB) GO TO 255                                                 TFA04540
      CEE=CRE+DX                                                        TFA04550
      CNN=CRN+DY                                                        TFA04560
  255 IF(LLR) GO TO 265                                                 TFA04570
C                                                                       TFA04580
C         LAYER LIMITS TEST.                                            TFA04590
      IF(KLR.EQ.KMAX) GO TO 260                                         TFA04600
      IF(DEP2.GT.ZC(KLW,ISSP)) GO TO 450                                TFA04610
  260 IF(KLR.EQ.1) GO TO 265                                            TFA04620
      IF(DEP2.LT.ZC(KUP,ISSP)) GO TO 440                                TFA04630
  265 IF(LBE) GO TO 300                                                 TFA04640
      IF(LHORZ) GO TO 280                                               TFA04650
C                                                                       TFA04660
C         CHANNEL LIMITS TEST.  CHECK FOR RAY TURNAROUND OR REFLECTION. TFA04670
C         DEP IS THE BACKUP POINT.                                      TFA04680
      IF(.NOT.LRFR.OR.(TH2.NE.0.)) GO TO 270                            TFA04690
      DEP=CHS                                                           TFA04700
      IF(DZ.GT.0.) DEP=CHD                                              TFA04710
      GO TO 420                                                         TFA04720
  270 IF(DEP2.LT.CHS) GO TO 410                                         TFA04730
      IF(DEP2.GT.CHD) GO TO 415                                         TFA04740
C                                                                       TFA04750
C         TEST FOR BOTTOM CONTACT.                                      TFA04760
  280 WH2=DEEP(CEE,CNN,DHN,ZB,IER)                                      TFA04770
      IF(IER.NE.0) GO TO 520                                            TFA04780
      IF(DEP2.GE.WH2) GO TO 390                                         TFA04790
C                                                                       TFA04800
C         ADVANCE THE TRACE POSITION, TAIL TO HEAD.                     TFA04810
  290 HDIS=HDIS+DR                                                      TFA04820
  300 CRE=CEE                                                           TFA04830
```

```
      CRN=CNN                                                           TFA04840
      DEP1=DEP2                                                         TFA04850
      WH1=WH2                                                           TFA04860
      NSEG=NSEG+1                                                       TFA04870
C                                                                       TFA04880
C         TIME CALCULATION.                                             TFA04890
  310 IF(LREFR) GO TO 320                                               TFA04900
      DT=DLF/C1                                                         TFA04910
      GO TO 330                                                         TFA04920
  320 C2=C1+G*DZ                                                        TFA04930
      C1=C2                                                             TFA04940
      W1=W2                                                             TFA04950
      W2=DZ+C2/G                                                        TFA04960
      DT=DLOG(W2*(1.+ST1)/(W1*(1.+ST2)))/G                              TFA04970
  330 ELAPS=ELAPS+DT                                                    TFA04980
C                                                                       TFA04990
C         IF AT A REVERSAL POINT, A DIRECTION CHANGE IS NECESSARY.      TFA05000
  340 IF(LBB) GO TO 380                                                 TFA05010
      IF(LHORZ) GO TO 360                                               TFA05020
      IF(LREV) GO TO 370                                                TFA05030
C                                                                       TFA05040
C         NO REVERSAL.                                                  TFA05050
      CT1=CT2                                                           TFA05060
      ST1=ST2                                                           TFA05070
      TH1=TH2                                                           TFA05080
      CTYPE=CCHCK                                                       TFA05090
      GO TO 160                                                         TFA05100
C                                                                       TFA05110
C         TURNAROUND.                                                   TFA05120
  360 CT1=1.                                                            TFA05130
      ST1=0.                                                            TFA05140
      TH1=0.                                                            TFA05150
      CTYPE=COVER                                                       TFA05160
      IF(DZ.GT.0.)CTYPE=CUNDR                                           TFA05170
      GO TO 160                                                         TFA05180
C                                                                       TFA05190
C         SURFACE REFLECTION.                                           TFA05200
  370 ST1=-ST2                                                          TFA05210
      TH1=-TH2                                                          TFA05220
      CT1=CT2                                                           TFA05230
      CTYPE=CSURF                                                       TFA05240
      GO TO 160                                                         TFA05250
C                                                                       TFA05260
C         BOTTOM BOUNCE CHANGES RAY PARAMETERS.                         TFA05270
  380 NBNC=NBNC+1                                                       TFA05280
      FBOT(IFC,NBNC)=DEP2*1000.                                         TFA05290
      ELEV = TH1/DETRA                                                  TFA05300
      FANGLE(IFC,NBNC)=ELEV                                             TFA05310
```

```
      CALL ECANG(PHI,TH1,CT1,ST1,DHN,UVX,UVY,GRAZ,SLOPE,ZB)
      AHDG = PHI/DETRA
      IF(AHDG.LE.0.)AHDG=AHDG+360.
      FRAZ=GRAZ/DETRA
      BANGLE=SLOPE/DETRA
      IF(LOSS) BTMLSS=BTMLSS+FBLOSS(FRAZ)
      CO=C1
      CTYPE=CBOTT
      NPTS=NPTS+1
      GO TO 150
C
C         BOTTOM CONTACT.  BACKTRACK TO BOTTOM DEPTH.
  390 LREV=.TRUE.
      LBB=.TRUE.
      LHORZ=.FALSE.
      LLR=.FALSE.
      IF(DEP2.EQ.WH2) GO TO 250
      CALL CONTAC(CRE,CRN,DEP1,WH1,CEE,CNN,DEP2,WH2,UVX,UVY,
     1   TH1,DR,DHN,ZB,NLPS,&650)
      DEP=DEP2
      GO TO 425
C
C         CHANNEL LIMIT REACHED; FORCE A REVERSAL.
  410 DEP=CHS
      GO TO 420
  415 DEP=CHD
  420 LREV=.TRUE.
      LBB=.FALSE.
      LLR=.FALSE.
      LHORZ=DEP.GT.0.
  425 IF(DEP1.EQ.DEP) GO TO 340
  426 DEP2=DEP
      DZ=DEP2-DEP1
      IF(LHORZ) GO TO 230
      IF(LRFR) GO TO 430
      DLP=DZ/ST1
      GO TO 220
C
C         BECAUSE OF REFRACTION, RAY HEAD ELEVATION ANGLE CHANGES
C         AFTER DEPTH CHANGE.
  430 CT2=CT1-DZ/RHO
      IF(CT2.GT.1.) GO TO 230
      TH2=DSIGN(DARCOS(CT2),DZ)
      ST2=DSIN(TH2)
      GO TO 245
C
C         RAY PENETRATES NEXT SHALLOWER LAYER.
  440 DEP=ZC(KLP,ISSP)
```

```
      KUP=KLP-1                                                          TFA05800
      KLW=KLW-1                                                          TFA05810
      GO TO 460                                                          TFA05820
C                                                                        TFA05830
C        RAY PENETRATES NEXT DEEPER LAYER.                               TFA05840
  450 DEP=ZC(KLW,ISSP)                                                   TFA05850
      KLP=KLP+1                                                          TFA05860
      KLW=KLW+1                                                          TFA05870
  460 LLR=.TRUE.                                                         TFA05880
      LHORZ=.FALSE.                                                      TFA05890
      LREV=.FALSE.                                                       TFA05900
      LBE=.FALSE.                                                        TFA05910
      IF(DEP1.EQ.DEP) GO TO 164                                          TFA05920
      GO TO 426                                                          TFA05930
C                                                                        TFA05940
C        TRANSFER TO THE FOLLOWING SECTION ABORTS THE RAYTRACE.          TFA05950
  500 PRINT 6260                                                         TFA05960
 6260 FORMAT(/,10X,'ERROR AT BOTTOM CONTACT,')                           TFA05970
      GO TO 560                                                          TFA05980
  510 PRINT 6270                                                         TFA05990
 6270 FORMAT(/10X,'BOTTOM LOSS GATE EXCEEDED,')                          TFA06000
      GO TO 560                                                          TFA06010
  520 PRINT 6280                                                         TFA06020
 6280 FORMAT(/10X,'RAY LEFT GRID,')                                      TFA06030
      GO TO 560                                                          TFA06040
  530 PRINT 6290, WH2,HSTOP                                              TFA06050
 6290 FORMAT(/10X,'BOTTOM DEPTH OF ',F7.3,' KM CUT OF ',F7.3,' TO 10.0 KTFA06060
     1M RANGE,')                                                         TFA06070
      GO TO 560                                                          TFA06080
  540 PRINT 6300                                                         TFA06090
 6300 FORMAT(/10X,'TRACE DATA EXCEEDS OUTPUT PARAMETERS,')               TFA06100
      GO TO 560                                                          TFA06110
  550 PRINT 6310                                                         TFA06120
 6310 FORMAT(/10X,'RAY BECAME TRAPPED IN DUCT OR SOUND CHANNEL,')        TFA06130
  560 CONTINUE                                                           TFA06140
      PUNCH 7000,ELEVO,AHDGO,NPTS,KRAY                                   TFA06150
 7000 FORMAT(2F10.3,2I5)                                                 TFA06160
      PUNCH 7020,(XPUN(I),YPUN(I),I=1,NPTS)                              TFA06170
      PRINT 6320,NSEG                                                    TFA06180
 6320 FORMAT(10X,'TERMINATING RAYTRACE AFTER ',I5,' SEGMENTS.')          TFA06190
      PRINT 6330, NCTPL,NREV,NENC                                        TFA06200
 6330 FORMAT(10X,'PLOT BASED ON ',I5,' POINTS,',I5,  ' REVERSALS AND '   TFA06210
     1,I5,' BOTTOM BOUNCES.')                                            TFA06220
      IF(.NOT.LPLOT) GO TO 570                                           TFA06230
C                                                                        TFA06240
C        PLOT RAY'S HORIZONTAL TRACK.                                    TFA06250
      CALL T2DPLT(XPUN,YPUN,XMIN,YMIN,FDV,NPTS,IMK,KRAY)                 TFA06260
  570 IF(LNOR) GO TO 575                                                 TFA06270
```

```
      PRINT 6340,CRMIN,CTIME
 6340 FORMAT( 10X,'MINIMUM DISTANCE TO RECEIVER, ',F8.2,' KM, OCCURS AT
     1TIME ',F10.3/)
      NRB(KRAY)=NBNC
      CBE(KRAY)=CTHD
      CTT(KRAY)=CTIME
      CRM(KRAY)=CRMIN
  575 CTYPE=CSTOP
      EL=TH1/DETRA
      PRINT 5040,NREV,OTYPE,CRE,CRN,DEP1,AHDG,EL,BTR,DIST,ELAPS,
     1   CMAS(ISSF),KUP,CHS,CHD,C1
      NBNCF=MAX0(NBNCF,NENC)
      IF(IHD.LT.NHD) GO TO 130
C
C        RAY TRACES FOR RAYFAN COMPLETE.
  580 IF(NENCF.EQ.0) GO TO 610
      NTIMES=NHD/15
      NLEFT=NHD-15*NTIMES
      PRINT 6350,IEL,ELEVO
 6350 FORMAT('1',9X,'RAYFAN',I3,' COMPLETE.  INITIAL ELEVATION ANGLE = '
     1,F8.3,' DEGREES.'/)
      IF(NTIMES.EQ.0) GO TO 600
      DO 590 J=1,NTIMES
      KS=(J-1)*15+1
      KE=KS+14
      CALL ANGPRT(KS,KE,NBNCF,CHD,1)
      CALL ANGPRT(KS,KE,NBNCF,CHD,2)
  590 CONTINUE
  600 IF(NLEFT.EQ.0)  GO TO 610
      KS=15*NTIMES+1
      KE=NHD
      CALL ANGPRT(KS,KE,NBNCF,CHD,1)
      CALL ANGPRT(KS,KE,NBNCF,CHD,2)
  610 IF(IEL.LT.NEL) GO TO 110
C
C         ALL TRACES COMPLETE. OUTPUT RAY SUMMARY.
      PUNCH 7020,FEND
      PRINT 6360,KRAY
      IF(LNOR) GO TO 640
      PRINT 6370
      PRINT 6380
 6360 FORMAT('1',I4,' RAY TRACES COMPLETED.'//10X,
     1'D  MINIMUM DISTANCE TO RECEIVER'/10X,'H  HEADING AT CLOSEST APPRO
     2ACH TO RECEIVER'/10X,'T  TIME AT CLOSEST APPROACH TO RECEIVER')
 6370 FORMAT(/,16X,'RAY',3X,'INITIAL',3X,'INITIAL',1X,'NUMBER OF',5X,
     1 'H',9X,'T',9X,'D')
 6380 FORMAT(13X,'NUMBER',3X,'HEADING',1X,'ELEVATION',3X,'BOUNCES')
      DO 630 J=1,KRAY
```

```
      PRINT 6390,J,OHD(J),CEL(J),NRB(J),OBB(J),CTT(J),ORM(J)         TFA06760
 6390 FORMAT(16X,I3,2F10.3,7X,I3,3F10.3)                              TFA06770
  630 CONTINUE                                                        TFA06780
  640 IF(.NOT.LFLOT) RETURN                                           TFA06790
C                                                                     TFA06800
C        PLOT LEGEND, THEN LEAVE PLOT ENVIRONMENT.                    TFA06810
      CALL ENDFLT(SNGL(DIRA),ELST,ELDV,HDWID,NEL)                     TFA06820
      RETURN                                                          TFA06830
      END                                                             TFA06840
C                                                                     TFA06850
C                                                                     TFA06860
C       SUBROUTINE GRDSCT MAY BE FOUND IN THE G3DF APPENDIX LISTING   TFA06870
C                                                                     TFA06880
C                                                                     TFA06890
```

```
C
C
C
C
C
C
C     SUBROUTINES FOLLOW:
C                          IDFPOF
C                          NUPROF
C                          CHNLIM
C                          GRDSCT
C
C*********************************************************************
C
C
      SUBROUTINE IDFPOF(LNUF)
C
C
C        DETERMINES WATER MASS INDEX BASED ON FRONTAL PENETRATION.
C
      IMPLICIT REAL*8(C-D),REAL*8(G),REAL*8(P-W)
      IMPLICIT LOGICAL*4(L)
      COMMON /PROFIL/OMAS(3),LNOT(2),LEW(2),LNS(2),FENTY(2),FENTX(2),
     1   FM(2)
      COMMON/CHANEL/CO,CRE,CRN,VV,DEP1,CHS,CHD,
     1   ZMAX,LCAL(3),NVAL(3),KUP,KLW,ISSP,KMAX
      X=CRE
      Y=CRN
      IPRO=ISSP
      ISSP = 1
      DO 1 I=1,2
      IF(LNOT(I)) GO TO 4
      IF(LEW(I)) GO TO 2
      IF(LNS(I)) GO TO 3
      EBND=(Y-FENTY(I))/FM(I)+FENTX(I)
      IF(X.LT.EBND) GO TO 4
      GO TO 1
    2 IF(Y.LT.FENTY(I)) GO TO 4
      GO TO 1
    3 IF(X.LT.FENTX(I)) GO TO 4
    1 ISSP = ISSP+1
    4 LNUP=IPRO.NE.ISSP
      KMAX=NVAL(ISSP)
      IF(LNUF) CALL NUPROF
      RETURN
      END
C
C
      SUBROUTINE NUPROF
```

```
C
C
C
C      CALCULATES NEW WATER MASS PARAMETERS.
C
      IMPLICIT REAL*8(C-D),REAL*8(G),REAL*8(P-W)
     COMMON/PARAM/C(50,3),GRADV(50,3),PIQT,ZC(50,3)
     COMMON/CHANEL/CO,CRE,CRN,VV,DEP1,CHS,CHD,
    1   ZMMX,LCAL(3),NVAL(3),KUP,KLW,ISSP,KMAX
     LOGICAL LCAL
     IF(LCAL(ISSP)) GO TO 15
     DO 10 K=1,KMAX
        C(K,ISSP)=C(K,ISSP)/1000.
  10    ZC(K,ISSP)=ZC(K,ISSP)/1000.
     LCAL(ISSP)=.TRUE.
  15 CONTINUE
C
C       IDENTIFY LAYER
     DO 20 K=1,KMAX
        IF(DEP1.LT.ZC(K,ISSP)) GO TO 30
  20    CONTINUE
     KM1=K
     GO TO 31
  30 KM1=K-1
  31 CO=C(KM1,ISSP)+ (DEP1-ZC(KM1,ISSP))*GRADV(KM1,ISSP)
     KUP=KM1
     KLW=KUP+1
     RETURN
     END
C
C
     SUBROUTINE CHNLIM
C
C
C
C       SOURCE:  L. R. FRAZO
C       PURPOSE: CALCULATES REFRACTIVE SOUND CHANNEL LIMITS BASED ON A
C       NEW SNELL'S LAW CONSTANT.  ABSOLUTE LIMITS SURFACE TO 10 KM.
      IMPLICIT REAL*8(C-D),REAL*8(G),REAL*8(P-W)
      IMPLICIT LOGICAL*4(L)
     COMMON/PARAM/C(50,3),GRADV(50,3),PIQT,ZC(50,3)
     COMMON/CHANEL/CO,CRE,CRN,VV,DEP1,CHS,CHD,
    1   ZMMX,LCAL(3),NVAL(3),KUP,KLW,ISSP,KMAX
      GRAD=GRADV(KUP,ISSP)
      LREF=GRAD.NE.0.
      LVER=VV.EQ.CO
      IF(.NOT.LVER) GO TO 10
      LNEG=GRAD.LE.0.
      LPOS=GRAD.GE.0.
```

```
      IF(LPOS) CHS=DEP1
      IF(LNEG) CHD=DEP1
      IF(.NOT.LRFR) RETURN
      IF(LPOS) GO TO 61
C
C        CALCULATE DEEP LIMIT.
   10 LARGE=VV.GT.1.6
      IF(LARGE) GO TO 55
      IF(KLP.LT.KMAX) GO TO 20
      IF(GRAD) 55,55,50
   20 IF(C(KLW,ISSP).GE.VV.AND.LRFR) GO TO 50
      IF(KLW.EQ.KMAX) GO TO 55
      KSTART = KLW+1
      DO 30 K=KSTART,KMAX
      KM1=K-1
      IF(C(K,ISSP).GE.VV.AND.GRADV(KM1,ISSP).NE.0.) GO TO 40
   30 CONTINUE
      IF(GRADV(KM1,ISSP)) 55,55,35
   35 KM1=K
   40 CHD=ZC(KM1,ISSP)+(VV-C(KM1,ISSP))/GRADV(KM1,ISSP)
      GO TO 60
   50 CHD=DEP1+(VV-CO)/GRAD
      GO TO 60
   55 CHD=10.
      IF(LARGE) GO TO 100
C
C        CALCULATE SHALLOW LIMIT.
   60 IF(LVER) RETURN
   61 IF(C(KLP,ISSP).GE.VV.AND.LRFR) GO TO 80
      IF(KLP.EQ.1) GO TO 100
      KEND=KLP-1
      DO 70 K=1,KEND
      KSTP=KLP-K
      IF(C(KSTP,ISSP).GE.VV.AND.GRADV(KSTP,ISSP).NE.0.) GO TO 90
   70 CONTINUE
      GO TO 100
   80 CHS=DEP1+(VV-CO)/GRAD
      RETURN
   90 CHS=ZC(KSTP,ISSP)+(VV-C(KSTP,ISSP))/GRADV(KSTP,ISSP)
      RETURN
  100 CHS=0.
      RETURN
      END
C
C
      SUBROUTINE GRDSCT(XST,YST,XMAX,YMAX,IMAX,JMAX,DHN,ZB)
C
C
```

```
      REAL*8 CHN
      COMMON /DIMS/ KMX,KMY
      DIMENSION ZB(KMX,KMY), D10(10)
      CHN=1.
      XST=0.
      YST=0.
      XMAX=149.
      YMAX=149.
      IMAX=150
      JMAX=150
      READ(1,1000) ZB
 1000 FORMAT(10F8.4)
      RETURN
      END
```

```
C
C
C
C
C
C     SUBROUTINES FOLLOW:
C                               ICTSUB
C                               DEEP
C                               CONTAC
C                               ECANC
C                               ANGPRT
C                               FBLOSS
C
C
C**************************************************************************
C
C
      SUBROUTINE ICTSUB(X,Y,DHN,IER)
C
C
C        SOURCE:  GEORGE GIELIS, NRL
C        PURPOSE: IDS BALLPARK SUBSCRIPTS UNDER HORIZONTAL COORDINATES.
      REAL*8 DX,DY,DHN,X,Y,XX,YY
      COMMON/SUBS/DX,DY,ILT,IRT,IVR,JBT,JTP,JVR,XST,YST
      COMMON /DIMS/ KMX,KMY,KITR
      XX=X-XST
      YY=Y-YST
      ILT=XX/DHN+1.0
      IRT=ILT+1
      JBT=YY/DHN+1.0
      JTP=JBT+1
      XX=XX-(ILT-1)*DHN
      YY=YY-(JBT-1)*DHN
      IF (XX.LT.YY) GO TO 10
      IVR=IRT
      JVR=JBT
      GO TO 20
   10 CONTINUE
      IVR=ILT
      JVR=JTP
   20 IER=0
      IF(ILT.LT.1) IER=IER-1
      IF(JBT.LT.1) IER=IER-2
      IF(IRT.GT.KMX) IER=IER+1
      IF(JTP.GT.KMY) IER=IER+2
      RETURN
      END
C
```

```
C                                                                        NFL00520
      FUNCTION DEEP(CRE,CRN,DHN,ZB,IER)                                  NFL00530
C                                                                        NFL00540
C                                                                        NFL00550
C        SOURCE:  GEORGE GIELIS, NRL                                     NFL00560
C        PURPOSE: GETS WATER COLUMN DEPTH UNDER HORIZONTAL COORDINATES.  NFL00570
      IMPLICIT REAL*8(C-D),REAL*8(G),REAL*8(P-W)                         NFL00580
      COMMON /DIMS/ KMX,KMY,KITR                                         NFL00590
      COMMON/SUBS/DX,DY,ILT,IRT,IVR,JBT,JTP,JVR,XST,YST                  NFL00600
      DIMENSION ZB(KMX,KMY)                                              NFL00610
      CALL IDTSUB(CRE,CRN,DHN,IER)                                       NFL00620
      IF(IER.NE.0) RETURN                                                NFL00630
      CX=CRE-(ILT-1)*DHN-XST                                             NFL00640
      CY=CRN-(JBT-1)*DHN-YST                                             NFL00650
      D1=ZB(ILT,JBT)                                                     NFL00660
      D2=ZB(IRT,JTP)                                                     NFL00670
      DV=ZB(IVR,JVR)                                                     NFL00680
      IF(JVR.EQ.JTP) GO TO 10                                            NFL00690
      DX=(DV-D1)/DHN                                                     NFL00700
      DY=(D2-DV)/DHN                                                     NFL00710
      GO TO 20                                                           NFL00720
   10 CONTINUE                                                           NFL00730
      DX=(D2-DV)/DHN                                                     NFL00740
      DY=(DV-D1)/DHN                                                     NFL00750
   20 CONTINUE                                                           NFL00760
      DEEP=DX*CX+DY*CY+D1                                                NFL00770
      RETURN                                                             NFL00780
      END                                                                NFL00790
C                                                                        NFL00800
C                                                                        NFL00810
      SUBROUTINE CONTAC(CRE,CRN,DEP1,WH1,X2,Y2,Z2,H2,QRE,QRN,            NFL00820
C                                                                        NFL00830
C                                                                        NFL00840
     1   THETA,DELR,DHN,ZB,NLPS,*)                                       NFL00850
C        SOURCE:    L. R. ERAZO                                          NFL00860
C        PURPOSE: GETS 3D COORDINATES AT BOTTOM CONTACT POINT.           NFL00870
      IMPLICIT REAL*8(A-E),REAL*8(G-H),REAL*8(P-Z)                       NFL00880
      REAL*4 ZB                                                          NFL00890
      COMMON /DIMS/ KMX,KMY,KITR                                         NFL00900
      COMMON/SUBS/DX,DY,ILT,IRT,IVR,JBT,JTP,JVR                          NFL00910
      DIMENSION ZB(KMX,KMY)                                              NFL00920
C        INITIAL LOOK STEP HALF OF SEGMENT LENGTH.                       NFL00930
      DMOVE=DELR/2.                                                      NFL00940
      TTT=DTAN(THETA)                                                    NFL00950
      NLPS=0                                                             NFL00960
      X1=CRE                                                             NFL00970
      Y1=CRN                                                             NFL00980
      Z1=DEP1                                                            NFL00990
```

```
      H1=WH1                                                            NFL01000
      IF(Z1.EQ.H1) RETURN                                               NFL01010
      IF(Z1.LT.H1) GO TO 5                                              NFL01020
      PRINT 6000,X1,Y1,Z1,H1,X2,Y2,Z2,H2,DELR,DZ,DMOVE,NLPS             NFL01030
 6000 FORMAT(1X,11F10.5,I5)                                             NFL01040
      X1=X1-DMOVE*GRE                                                   NFL01050
      Y1=Y1-DMOVE*GRN                                                   NFL01060
C         RAY HEAD'S CELL SUBSCRIPTS.                                   NFL01070
    5 I2VR=IVR                                                          NFL01080
      J2VR=JVR                                                          NFL01090
      J2BT=JBT                                                          NFL01100
C         COLLECT SUBSCRIPTS AT RAY TAIL.                               NFL01110
      H1=DEEP(X1,Y1,DHN,ZB,IER)                                         NFL01120
C         RAY TAIL'S CELL.                                              NFL01130
   10 I1VR=IVR                                                          NFL01140
      J1VR=JVR                                                          NFL01150
      J1BT=JBT                                                          NFL01160
   20 IF(I1VR.NE.I2VR.OR.J1VR.NE.J2VR) GO TO 30                         NFL01170
      IF(J1BT.EQ.J2BT) GO TO 200                                        NFL01180
C          RAY TAIL IS NOT IN HEAD'S CELL.                              NFL01190
   30 NLPS=NLPS+1                                                       NFL01200
      IF(NLPS.GT.50) GO TO 230                                          NFL01210
C         TAKE ONE STEP FROM RAY TAIL.  SUBSCRIPTS APPLY TO STEP POINT. NFL01220
      XT=X1+DMOVE*GRE                                                   NFL01230
      YT=Y1+DMOVE*GRN                                                   NFL01240
      ZT=Z1+DMOVE*TTT                                                   NFL01250
      HT=DEEP(XT,YT,DHN,ZB,IER)                                         NFL01260
      IF(ZT.GE.HT) GO TO 40                                             NFL01270
C         STEP POINT IS ABOVE BOTTOM.  ADVANCE RAY TAIL TO HERE.        NFL01280
      X1=XT                                                             NFL01290
      Y1=YT                                                             NFL01300
      Z1=ZT                                                             NFL01310
      H1=HT                                                             NFL01320
      GO TO 10                                                          NFL01330
C         STEPPED INTO BOTTOM.  MOVE RAY HEAD TO HERE.                  NFL01340
   40 X2=XT                                                             NFL01350
      Y2=YT                                                             NFL01360
      Z2=ZT                                                             NFL01370
      H2=HT                                                             NFL01380
C         RAY HEAD'S NEW CELL.  HALVE THE LOOK STEP.                    NFL01390
      I2VR=IVR                                                          NFL01400
      J2VR=JVR                                                          NFL01410
      J2BT=JBT                                                          NFL01420
      DMOVE=DMOVE/2                                                     NFL01430
      GO TO 20                                                          NFL01440
C         ENTIRE SEGMENT IN SAME CELL.  MOVE RAY HEAD TO CONTACT POINT. NFL01450
  200 IF(Z2.EQ.H2) RETURN                                               NFL01460
      DZ=Z2-Z1                                                          NFL01470
```

```
      DR=DSQRT((X2-X1)**2+(Y2-Y1)**2)                                   NRL01480
      ZC=(H1*Z2-H2*Z1)/(DZ+H1-H2)                                       NRL01490
      IF(DZ.NE.0.) GO TO 210                                            NRL01500
      DMOVE=(Z1-H1)/(H1-H2)                                             NRL01510
      GO TO 220                                                         NRL01520
  210 DMOVE=(ZC-Z1)/DZ                                                  NRL01530
  220 DMOVE=DMOVE*DR                                                    NRL01540
      X2=X1+DMOVE*GRE                                                   NRL01550
      Y2=Y1+DMOVE*GRN                                                   NRL01560
      H2=DEEP(X2,Y2,DHN,ZB,IER)                                         NRL01570
      Z2=H2                                                             NRL01580
      RETURN                                                            NRL01590
  230 PRINT 6000,X1,Y1,Z1,H1,X2,Y2,Z2,H2,DR,DZ,DMOVE,NLPS               NRL01600
      RETURN 1                                                          NRL01610
      END                                                               NRL01620
C                                                                       NRL01630
C                                                                       NRL01640
      SUBROUTINE BCANG(PHI,THETA,COST,SINT,DHN,UVX,UVY,GRAZ,SLOPE,ZB)   NRL01650
C                                                                       NRL01660
C                                                                       NRL01670
C        SOURCE:  GEORGE GIELIS, L. R. BRAZC                            NRL01680
C        PURPOSE: CALCULATES NEW RAY DIRECTION AFTER BOTTOM CONTACT.    NRL01690
      IMPLICIT REAL*8(C-D),REAL*8(G),REAL*8(P-W)                        NRL01700
      COMMON /DIMS/ KMX,KMY,KITR                                        NRL01710
      COMMON/PARAM/C(50,3),GRADV(50,3),PIOT,ZC(50,3)                    NRL01720
      COMMON/SLBS/DX,DY,ILT,IRT,IVR,JBT,JTP,JVR                         NRL01730
      DIMENSION ZE(KMX,KMY)                                             NRL01740
C        V IS UNIT VECTOR IN RAY DIRECTION.                             NRL01750
      UVR=COST                                                          NRL01760
      VX=UVX*UVR                                                        NRL01770
      VY=UVY*UVR                                                        NRL01780
      VZ=SINT                                                           NRL01790
C        UNIT VECTOR NORMAL TO TRIANGULAR FACET AT BOUNCE POINT.  DX, DYNRL01800
C        ARE PARTIAL DERIVATIVES OF Z W/R TO AXES (RATE OF DEPTH CHANGE NRL01810
C        ALONG  AXES) CALCULATED IN DEEP.  DZ IS MADE NEGATIVE TO PRO-  NRL01820
C        JECT THE NORMAL UP FROM THE BOTTOM.                            NRL01830
      DZ=-1.0                                                           NRL01840
      FNL=DSQRT(DX**2+DY**2+DZ**2)                                      NRL01850
      FNX=DX/FNL                                                        NRL01860
      PNY=DY/FNL                                                        NRL01870
      FNZ=DZ/FNL                                                        NRL01880
C        THE SLOPE OF THE FACET IS THE ANGLE THE UNIT NORMAL MAKES      NRL01890
C        WITH THE Z AXIS.                                               NRL01900
      SLOPE=DARCOS(FNZ)                                                 NRL01910
      DOT=VX*FNX+VY*PNY+VZ*FNZ                                          NRL01920
      GRAZ=DARSIN(-DOT)                                                 NRL01930
      CON=-2*DOT                                                        NRL01940
      VPX=VX+CON*FNX                                                    NRL01950
```

```
      VPY=VY+CCN*FNY
      VPZ=VZ+CCN*FNZ
      THETA=DARSIN(VPZ)
      VPR=DCOS(THETA)
      SINT=VPX/VPR
      COST=VPY/VPR
      UVX=SINT
      UVY=COST
      PHI=DATAN2(SINT,COST)
      RETURN
      END
C
C
      SUBROUTINE ANGPRT(LS,LE,NBOUNC,HEAD,ITYPE)
C
C
C        SOURCE:  GEORGE GIELIS, NRL
C        REVISED: L. R. ERAZO, FEB  1980
C        PURPOSE: OUTPUT OF BOUNCE DATA.
      COMMON/PRINTS/FECT(30,50),FANGLE(30,50)
      DIMENSION HEAD(100),KB(50)
      IF(ITYPE.EQ.2) GO TO 10
      PRINT 1400
 1400 FORMAT(/10X,'ELEVATION ANGLES IN DEGREES BEFORE BOUNCE:')
      GO TO 15
   10 PRINT 1500
 1500 FORMAT(/10X,'DEPTH IN METERS AT BOUNCE:')
   15 PRINT 1100,(HEAD(I),I=LS,LE)
 1100 FORMAT(/T5,'HEADINGS:',15F8.2)
      PRINT 1011
 1011 FORMAT(3X,'BOUNCE')
      DO 20 I=1,NBOUNC
      KB(I)=I
   20 CONTINUE
      IF(ITYPE.EQ.2)  GO TO 40
      DO 30 I=1,NBOUNC
      PRINT 1200,KB(I),((FANGLE(J,K),J=LS,LE),K=I,I)
 1200 FORMAT(1X,I6,6X,15F8.2)
   30 CONTINUE
      RETURN
   40 CONTINUE
      DO 45 I=1,NBOUNC
      PRINT 1300,KB(I),((FECT(J,K),J=LS,LE),K=I,I)
 1300 FORMAT(1X,I6,6X,15F8.1)
   45 CONTINUE
      RETURN
      END
C
```

```
C                                                                          NFL02440
      FUNCTION FBLOSS(ANGLE)                                               NFL02450
C                                                                          NFL02460
C                                                                          NFL02470
      DIMENSION ENDS(8),ELOSS(8)                                           NFL02480
      DATA ENDS/0.0,11.0,20.0,25.0,35.0,45.0,56.0,90.0/                    NFL02490
      DATA ELOSS/0.0,0.0,3.0,4.4,6.7,8.5,10.0,10.0/                        NFL02500
      DO 10 I=1,7                                                          NFL02510
      IF(ANGLE.GT.ENDS(I).AND.ANGLE.LE.ENDS(I+1)) GO TO 20                 NFL02520
   10 CONTINUE                                                             NFL02530
      IF(I.GT.7) GO TO 30                                                  NFL02540
   20 CONTINUE                                                             NFL02550
      SLOPE=(ELOSS(I+1)-ELOSS(I))/(ENDS(I+1)-ENDS(I))                      NFL02560
      FBLOSS=ELOSS(I)+SLOPE*(ANGLE-ENDS(I))                                NFL02570
      GO TO 40                                                             NFL02580
   30 FBLOSS=10.0                                                          NFL02590
   40 CONTINUE                                                             NFL02600
      RETURN                                                               NFL02610
      END                                                                  NFL02620
```

```
C
C
C     SUBROUTINES FOLLOW:
C
C                        BGNPLT
C                        BDTPLT
C                        SSPPLT
C                        RNGPLT
C                        T2DPLT
C                        ENDPLT
C
C
C*************************************************************************
C
C
      SUBROUTINE BGNPLT(CTIT,XST,YST,FDV,RST,RDV,ZST,ZDV,ZEND,
C
C
     1   XREC,YREC,RREC,ZREC,FACT)
      REAL*8 TTL(12),CNX,CNY,CNLAT,CNLON
      COMMON /PLOTIT/ TTL,CNX,CNY,CNLAT,CNLON
      DIMENSION CTIT(20)
      CALL WINDOW(0.,25.*FACT,0.,21.*FACT)
      CALL PLOTS(0,0,0)
      CALL FACTOR(FACT)
      CALL PLOT(0.,0.,-3)
      CALL PLOT(0.,20.95,2)
      CALL PLOT(24.95,20.95,2)
      CALL PLOT(24.95,0.,2)
      CALL PLOT(0.,0.,2)
      CALL SYMBOL(1.,20.6,.12,CTIT,0.,80)
C        DRAW HORIZONTAL PLOT BORDER.
      CALL PLOT(1.,1.,-3)
      CALL AXIS(0.,0.,'X (KM)',-6,15.,0.,XST,FDV)
      CALL PLOT(15.,0.,2)
      CALL AXIS(0.,0.,'Y (KM)',6,15.,90.,YST,FDV)
      CALL PLOT(0.,15.,2)
      CALL PLOT(15.,15.,2)
      CALL PLOT(15.,0.,2)
      CALL SYMBOL(0.,-.6,.12,TTL,0.,96)
      XSYM=(XREC-XST)/FDV
      YSYM=(YREC-YST)/FDV
      CALL SYMBOL(XSYM,YSYM,.10,'.REC',0.,4)
      XCEN=(CNX-XST)/FDV
      IF(XCEN.LT.0..OR.XCEN.GT.15.) GO TO 10
      YCEN=(CNY-YST)/FDV
      IF(YCEN.LT.0..OR.YCEN.GT.15.) GO TO 10
      CALL SYMBOL(XCEN,YCEN,.10,12,0.,-1)
```

```
      CALL NUMBER(999.,YCEN,.10,CNLAT,0.,5)
      CALL SYMBOL(999.,YCEN,.10,13,0.,-1)
      CALL NUMBER(999.,YCEN,.10,CNLON,0.,5)
C        DRAW VERTICAL PLOT BORDER.
   10 CALL PLOT(0.,15.5,-3)
      CALL AXIS(0.,3.5,'RANGE (KM)',10,19.,0.,RST,RDV)
      CALL PLOT(19.,3.5,2)
      CALL AXIS(0.,3.5,'DEPTH (KM)',-10,3.5,-90.,ZST,ZDV)
      CALL PLOT(0.,0.,2)
      CALL PLOT(19.,0.,2)
      CALL PLOT(19.,3.5,2)
      RSYM=(RREC-RST)/RDV
      ZSYM=(ZEND-ZST-ZREC)/ZDV
      CALL SYMBOL(RSYM,ZSYM,.10,'.REC',0.,4)
      RETURN
      END
C
C
      SUBROUTINE BOTPLT(DTRX,DTRY,DHN,WHO,XO,YO,ZEND,XMIN,XMAX,YMIN,
C
C
     1   YMAX,ZST,ZDV,ARST,ARDV,ZB)
      REAL*8 DHN,DTR,WHO,DTRX,DTRY,TRKX,TRKY
      COMMON /DIMS/ KMX,KMY
      DIMENSION RTRK(193),HTRK(193),IPAT(4),ZB(KMX,KMY)
      DATA IPAT/257,514,1028,2056/
      DTR=DSQRT(DTRX**2+DTRY**2)/ARDV
      HTRK(1)=0.
      RTRK(1)=0.
      TRKX=XO+ARST*DTRX
      TRKY=YO+ARST*DTRY
      TRKX=AMAX1(SNGL(TRKX),XMIN)
      TRKY=AMAX1(SNGL(TRKY),YMIN)
      TRKX=AMIN1(SNGL(TRKX),XMAX)
      TRKY=AMIN1(SNGL(TRKY),YMAX)
      RTRK(2)=0.
      DPTH=DEEP(TRKX,TRKY,DHN,ZB,IER)
      DPTH=AMAX1(DPTH,ZST)
      DPTH=AMIN1(DPTH,ZEND)
      HTRK(2)=(ZEND-DPTH)/ZDV
      DO 10 I=3,192
      IM1=I-1
      ISTOP=I
      TRKX=TRKX+DTRX
      IF(TRKX.GT.XMAX.OR.TRKX.LT.XMIN) GO TO 20
      TRKY=TRKY+DTRY
      IF(TRKY.GT.YMAX.OR.TRKY.LT.YMIN) GO TO 20
      RTR=RTRK(IM1)+DTR
```

```
      IF(RTR.GT.19.) GO TO 20                                           ECP01000
      RTRK(I)=RTR                                                       ECP01010
      DPTH=DEEP(TRKX,TRKY,DHN,ZB,IER)                                   ECP01020
      DPTH=AMAX1(DPTH,ZST)                                              ECP01030
      DPTH=AMIN1(DPTH,ZEND)                                             ECP01040
      HTRK(I)=(ZEND-DPTH)/ZDV                                           ECP01050
   10 CONTINUE                                                          ECP01060
   20 HTRK(ISTOP)=0.                                                    ECP01070
      RTRK(ISTOP)=RTRK(IM1)                                             ECP01080
      SURL=ZEND/ZDV                                                     ECP01090
      CALL PLOT(19.,SURL,3)                                             ECP01100
      CALL PLOT(0.,SURL,2)                                              ECP01110
      CALL TONE(0.,0.,IPAT,-4)                                          ECP01120
      CALL TONE(RTRK,HTRK,ISTOP,1)                                      ECP01130
      RETURN                                                            ECP01140
      END                                                               ECP01150
C                                                                       ECP01160
C                                                                       ECP01170
      SUBROUTINE SSPPLT(NVAL,NSSP,CMAS)                                 ECP01180
C                                                                       ECP01190
C                                                                       ECP01200
      REAL*8 C,GRADV,PIQT                                               ECP01210
      COMMON /PARAM/ C(50,3),GRADV(50,3),PIQT,ZC(50,3)                  ECP01220
      DIMENSION CP(52),ZCP(52),CMAS(NSSP),NVAL(NSSP)                    ECP01230
      XLOC=20.                                                          ECP01240
      YLOC=3.5                                                          ECP01250
      DO 10 I=1,3                                                       ECP01260
      CALL PLOT(XLOC,YLOC,-3)                                           ECP01270
      IF(I.GT.NSSP) GO TO 10                                            ECP01280
      CALL SYMBOL(0.,-.4,.12,CMAS(I),0.,4)                              ECP01290
      KMAX=NVAL(I)                                                      ECP01300
      DO 20 J=1,KMAX                                                    ECP01310
      CP(J)=C(J,I)                                                      ECP01320
   20 ZCP(J)=ZC(J,I)                                                    ECP01330
      CALL SCALE(CP,3.,KMAX,1)                                          ECP01340
      CALL SCALE(ZCP,6.,KMAX,1)                                         ECP01350
      KMP1=KMAX+1                                                       ECP01360
      KMP2=KMAX+2                                                       ECP01370
      CVAL=CP(KMP1)                                                     ECP01380
      CDV=CP(KMP2)                                                      ECP01390
      ZVAL=ZCP(KMP1)                                                    ECP01400
      ZDV=ZCP(KMP2)                                                     ECP01410
      CALL AXIS(0.,0.,'SOUND SPEED (M/S)',17,3.,0.,CVAL,CDV)            ECP01420
      CALL AXIS(0.,0.,'DEPTH (M)',-9,6.,-90.,ZVAL,ZDV)                  ECP01430
      DO 30 J=1,KMAX                                                    ECP01440
   30 ZCP(J)=-ZCP(J)                                                    ECP01450
      CALL LINE(CP,ZCP,KMAX,1,0,0)                                      ECP01460
      XLOC=0.                                                           ECP01470
```

```
1C YLCC=-7.
   CALL PLOT(-20.,10.5,-3)
   RETURN
   END
C
C
   SUBROUTINE RNGPLT(FRANGE,FDEPTH,DTYPE,LBB,IMK)
C
C
   LOGICAL LBB
   DATA CINIT/'INIT'/
   IPEN=2
   IF(DTYPE.EQ.CINIT) IPEN=3
   IF(FRANGE.LT.0.) IPEN=3
   IF(FRANGE.GT.19.) IPEN=3
   IF(FDEPTH.GT.3.5) IPEN=3
   IF(FDEPTH.LT.0.) IPEN=3
   CALL PLOT(FRANGE,FDEPTH,IPEN)
   IF(DTYPE.EQ.CINIT) GO TO 10
   IF(.NOT.LBB.OR.IPEN.EQ.3) RETURN
10 CALL SYMBOL(FRANGE,FDEPTH,.10,IMK,0.,-1)
   RETURN
   END
C
C
   SUBROUTINE T2DPLT(XFUN,YFUN,XST,YST,FDV,NPTS,IMK,KRAY)
C
C
   DIMENSION XFUN(103),YFUN(103)
   FPN=KRAY
   CALL PLOT(0.,-15.5,-3)
   NPP1=NPTS+1
   NPP2=NPTS+2
   XFUN(NPP1)=XST
   XFUN(NPP2)=FDV
   YFUN(NPP1)=YST
   YFUN(NPP2)=FDV
   CALL LINE(XFUN,YFUN,NPTS,1,1,IMK)
   XNOW=XFUN(NPTS)/FDV
   YNOW=YFUN(NPTS)/FDV
   CALL NUMBER(XNOW,YNOW,.10,FPN,0.,-1)
   CALL PLOT(0.,15.5,-3)
   RETURN
   END
C
C
   SUBROUTINE ENDPLT(DIRA,ELST,ELDV,HDWID,NFL)
C
```

```
C                                                                       ECP01960
      CALL PLOT(15.5,-0.6,-3)                                           ECP01970
      CALL SYMBOL(0.,0.,.12,'INITIAL ELEVATION ANGLES',0.,24)           ECP01980
      CALL SYMBOL(0.,-.25,.12,'RAYFAN   SYMBOL     THETA',0.,24)        ECP01990
      DO 10 I=1,NEL                                                     ECP02000
      INK=MOD(I-1,13)                                                   ECP02010
      YLOC=-.25*(I+1)                                                   ECP02020
      FPN=FLOAT(I)                                                      ECP02030
      CALL NUMBER(.4,YLOC,.12,FPN,0.,-1)                                ECP02040
      CALL SYMBOL(1.5,YLOC,.12,INK,0.,-1)                               ECP02050
      FPTHET=ELST+(I-1)*ELDV                                            ECP02060
   10 CALL NUMBER(2.3,YLOC,.12,FPTHET,0.,3)                             ECP02070
      YLOC=YLOC-.5                                                      ECP02080
      CALL SYMBOL(0.,YLOC,.12,'HEADINGS ',0.,9)                         ECP02090
      CALL NUMBER(999.,YLOC,0.,DIRA,0.,3)                               ECP02100
      CALL SYMBOL(999.,YLOC,0.,' +/- ',0.,5)                            ECP02110
      HAWID=HDWID/2.                                                    ECP02120
      CALL NUMBER(999.,YLOC,0.,HAWID,0.,3)                              ECP02130
      CALL PLOT(0.,0.,999)                                              ECP02140
      RETURN                                                            ECP02150
      END                                                               ECP02160
```

```
C
C
C
C                                   APPENDIX N
C
C
C*****************************************************************************
C
C
C                                     ECCOM
C
C        PROGRAM ECCOM IS USED TO COMBINE THE PUNCHED HISTORY FROM
C        ECTRACE RUNS TO ALLOW THE INVESTIGATION OF SHADOWING EFFECTS
C
C*****************************************************************************
C
//  EXEC  FORTCLGW
//FORT.SYSIN    DD  *
      DIMENSION XPUN(103),YPUN(103)                                       R2D00010
      REAL*8 TTL(12),CNX,CNY,CNLAT,CNLON                                  R2D00020
      DIMENSION CTIT(20)                                                  R2D00040
      READ 5021,CTIT,TTL,XST,YST,FDV,CNX,CNY,CNLAT,CNLON,FACT
C
C
C        IF THE USER DESIRES TO CHANGE SIZE OF THIS HORIZONTAL PLOT
C        FROM THAT OF THE FIRST ECTRACE RUN IN THE INPUT DECK, THE
C        SCALING FACTOR, 'FACT' MUST BE CHANGED. THIS VALUE IS THE LAST
C        NUMBER ON DATA CARD #4 OF OF THE FIRST ECTRACE PUNCHED HISTORY.
C        THE VALUE 1 RESULTS IN A 15 INCH PLOT.
C
C
      PRINT 6000,CTIT,TTL,XST,FDV,YST,FDV,CNX,CNY,CNLAT,CNLON,FACT
 6000 FORMAT(10X,20A4/10X,6A8/10X,6A8/4X,'XST',6X,'XDV',4X,'YST',5X,'YDV
     1',6X,'CNX',5X,'CNY',3X,'CNLAT',4X,'CNLON'/1X,8F8.3//10X,2('    INTI
     2AL')/10X,' ELEVATION   HEADING    POINTS',5X,'SCALE FACTOR=',F8.3)
      IF(FACT.LE.0.)FACT=1.
      SIZE=FACT*17.
      CALL WINDOW(0.,SIZE,0.,SIZE)
      CALL PLOTS(0,0,0)                                                   R2D00110
      CALL FACTOR(FACT)
      CALL PLOT(0.,0.,-3)                                                 R2D00120
      CALL PLOT(0.,17.,2)                                                 R2D00130
      CALL PLOT(17.,17.,2)                                                R2D00140
      CALL PLOT(17.,0.,2)                                                 R2D00150
      CALL PLOT(0.,0.,2)                                                  R2D00160
      CALL SYMBOL(1.,16.6,.12,CTIT,0.,80)                                 R2D00170
C
C        DRAW POSITION PLOT BORDER.                                       R2D00180
```

```
C
      CALL PLOT(1.,1.,-3)                                                R2D00190
      CALL AXIS(0.,0.,'X  (KM)',-6,15.,0.,XST,FDV)
      CALL PLOT(15.,0.,2)
      CALL AXIS(0.,0.,'Y  (KM)',6,15.,90.,YST,FDV)
      CALL PLOT(0.,15.,2)
      CALL PLOT(15.,15.,2)                                               R2D00220
      CALL PLOT(15.,0.,2)                                                R2D00230
      CALL SYMBOL(0.,-.6,.12,TTL,0.,96)                                  R2D00240
      XCEN=(CNX-XST)/FDV
      IF(XCEN.LT.0..OR.XCEN.GT.15.) GO TO 102
      YCEN=(CNY-YST)/FDV
      IF(YCEN.LT.0..OR.YCEN.GT.15.) GO TO 102
      CALL SYMBOL(XCEN,YCEN,.10,12,0.,-1)                                R2D00290
      CALL NUMBER(999.,YCEN,.10,CNLAT,0.,5)                              R2D00300
      CALL SYMBOL(999.,YCEN,.10,13,0.,-1)                                R2D00310
      CALL NUMBER(999.,YCEN,.10,CNLON,0.,5)                              R2D00320
      ELEVO=9999.                                                        R2D00330
      GO TO 102
C
C        PLOT ALL RAYS.
C
  100 READ(5,5021,END=200)TTIT,TTL,XST,YST,FDV,CNX,CNY,CNLAT,CNLON,FACT
 5021 FORMAT(20A4/6A8/6A8/10F8.3)                                        R2D00360
  102 ELEV1=ELEVO
      READ 5031,ELEVO,AHDGO,NPTS                                         R2D00390
 5031 FORMAT(2F10.3,I5)                                                  R2D00400
      IF (ELEVO.EQ.9999.) GO TO 100                                      R2D00420
      PRINT 5033,ELEVO,AHDGO,NPTS                                        R2D00430
 5033 FORMAT (8X,2F10.3,I9)
      READ 5041,(XPUN(I),YPUN(I),I=1,NPTS)                               R2D00450
 5041 FORMAT(10F8.3)                                                     R2D00460
      NPP1=NPTS+1
      NPP2=NPTS+2
      XPUN(NPP1)=XST
      XPUN(NPP2)=FDV
      YPUN(NPP1)=YST
      YPUN(NPP2)=FDV
      CALL LINE (XPUN,YPUN,NPTS,1,0,0)
      GO TO 102
C
C        END OF PLOT
C
  200 CALL PLOT(0.,0.,+999)
      STOP                                                               R2D00500
      END                                                                R2D00510
C
C        THE FOLLOWING CARDS ARE INSERTED BETWEEN THE PROGRAM LISTING
```

```
C          AND THE DATA.
C                /*
C                //GO.SYSIN DD *
C
C          FOLLOWING THIS CARD THE USER WILL PLACE THE ECTRACE TRIALS TO
C          BE COMBINED IN ONE ECCOM PLOT.
C
C          NO CARD CHANGES ARE NEEDED, HOWEVER IF THE USER DESIRES TO
C          HAVE THE PLOT TITLED ECCOM VICE ECTRACE , CHANGE THE FIRST
C          ECTRACE TITLE CARD TO 'ECCOM TRIAL' FROM 'ECTRACE TRIAL'.
C
C
C          AT THE END OF ECTRACE DATA HISTORY CARDS PLACE END OF JOB CARD
C
C*******************************************************************************
```

APPENDIX O
FORTCLGW JCL

The job control language (JCL) card for FORTCLGW used in several of the programs was

//EXEC FORTCLGW.REGION.GO=250K .

This JCL card identifies and calls the following JCL:

```
XXFORTCLGW PROC DEST=A,IMSL=SF
XXFORT  EXEC  PGM=IEYFORT,REGION=180K
XXSYSPRINT DD SYSOUT=&DEST,DCB=(RECFM=FBA,LRECL=120,BLKSIZE=3360)
SSSYSLIN DD UNIT=SYSDA,SPACE-(CYL,(1,1)),DISP=(,PASS),DSN=&&SYSLIN,
XX  DOE=(RECFM=FB,LRECL=80,BKJSUZE=800)
XXLINK  EXEC  PGM=IEWL,REGION=180K,PARM='MAP,LIST',COND=(4,LT)
XXSYSLIB DD  DISP=SHR,DSN=SYS1.FORTLIB
XX       DD  DISP=SHR,DSN=SYS1.MPSLIB
XX       DD  DISP=SHR,DSN=SYS3.IMSL.&IMSL
XX       DD  DISP=SHR,DSN=SYS1.VTECPLOT
XXSYSLMOD DD DSN=&T(PROGRAM),UNIT=SYSDA,DISP=(,PASS),
XX  SPACE=(CYL,(3,1,1))
XXSYSLIN  DD  DSN=*.FORT.SYSLIN,DISP=(OLD,DELETE)
XX       DD  DISP=SHR,DSN=SYS1.PROCLIE(VMAPP)
XX        DD  DDNAME=SYSIN
XXSYSPRINT DD   SYSOUT=&DEST,DOE=(RECFM=FBA,LRECL=121,BLKSIZE=1210),
XX  SPACE=(CYCL,(1,1)),UNIT=(SYSOUT,SEP=SYSLIN)
XXSYSUT1  DD  UNIT=(SYSDA,SEP=(SYSLIN,SYSLMOD,SYSLIB)),
XX  SPACE=(TRK,(10,5))
XXPLOT  EXEC PGM=IEVMAPP,COND=(4LT)
XXSTEPLIB DD  DSN=SYS1.VTECPLOT,DISP=SHR
XXPLOTLOG  DD  SYSOUT=A
XXVECTR1  DD  DISP=(OLD,DELETE),DSN=&&VECTR1
XXSYSVECTR  DD  DISP=(OLD,DELETE),DSN=&&VECTR2
XXSYSVECTOR  DD  SYSOUT=(A,,5555)
XXVECTTAPE  DD  DUMMY
```

LIST OF REFERENCES

1. Naval Research Laboratory Report 7815, Horizontal-Gradient Acoustic Ray-Trace Program TRIMAIN, by B. G. Roberts, Jr., 16 December 1974.

2. Harrison, C. H., "Three-Dimensional Ray Paths in Basins, Troughs, and Near Seamounts by Use of Ray Invariants," Journal of the Acoustical Society of America, v. 62, p. 1382-1388, December 1977.

3. Naval Research Laboratory Memorandum Report 3773, Bathymetric Effects on Three-Dimensional Ray Traces, by R. N. DeWitt, April 1978.

4. Naval Research Laboratory Chart, Bathymetry of the Norwegian Greenland and Western Barents Sea, compiled by R. K. Perry and others, 1977.

5. Naval Postgraduate School IBM Scientific Subroutine Package (SSP3), RAIN1 - a set of FORTRAN coded computer subroutines, programmed by R. Hilleary and L. Nolan, July 1974.

6. CONTUR, a FORTRAN-coded computer subroutine, Naval Postgraduate School non-ISML system library, programmed by Bernadette R. Peavey, July 1969.

7. Naval Postgraduate School Technical Note 0141-34, Using the Versatec Plotter at NPS, edited by Sharon D. Raney, December 1979.

8. Clay, C. S., and Medwin, H. Acoustical Oceanography: Principles and Applications, p. 83-87, Wiley, 1977.

9. Naval Oceanographic Office Technical Report 233, Synthetic Bathymetric Profiling Systen (SYNBAPS), by R. S. Van Wyckhouse, May 1973.

10. Naval Postgraduate School Technical Note 0141-08, Magnetic Tape Usage Under OS/MVP at NPS, by Sharon D. Raney, December 1977.

11. Naval Postgraduate School Technical Note 0141-05, User Libraries and Source Code Editing Under OS, by Sharon D. Raney, July 1977.

12. Naval Postgraduate School non-ISML system library, PTL3D1 - a FORTRAN coded computer subroutine, implemented at NPS by Sharon D. Raney, December 1973.

13. Harrison, C. H., "Acoustic Shadow Zones in the Horizontal Plane," Journal of the Acoustical Society of America, v. 65, p. 56-61, January 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department of Oceanography, Code 68M Naval Postgraduate School Monterey, California 93940	1
4.	Asst. Professor C. Dunlap, Code 68DU Department of Oceanography Naval Postgraduate School Monterey, California 93940	1
5.	Dr. G. Gielis Code 5123 Naval Research Laboratory Washington, D.C. 20390	1
6.	Mr. C. Votaw Code 5123 Naval Research Laboratory Washington, D.C. 20390	4
7.	LCDR R. Christianson Weapons Department (Code 331) Naval Postgraduate School Monterey, California 93940	1
8.	LT L. Erazo ASWOC Sigonella FPO NY 09253	1
9.	Oceanographer of the Navy Hoffman Building No. 2 200 Stovall Street Alexandria, Virginia 22332	1
10.	Commanding Officer Environmental Prediction Research Facility Monterey, California 93940	1
11.	Director Naval Research Laboratory Washington, D.C. 20390	1

12. Scientic Liaison Office, ONR 1
Scripps Institution of Oceanography
La Jolla, California 92037

13. Assoc. Professor A. Coppens 2
Code 61 Cz
Naval Postgraduate School
Monterey, California 93940

14. Assoc. Professor R. Shudde 1
Code 55 Su
Naval Postgraduate School
Monterey, California 93940

15. Department of Physics, Code 61 1
Naval Postgraduate School
Monterey, California 93940

16. Technical Library 1
Naval Underwater Systems Center
Newport, Rhode Island 02840

17. Technical Library 1
Naval Ocean Systems Center
San Diego, California 92152

18. Code (08T245) 1
Naval Material Department
Navy Department
Washington, D.C. 20360

19. Code (63R) 1
Naval Sea Systems Command
Washington, D.C. 20362

Thesis
C473 Christianson 189330
c.1
ECTRACE - Three dimensional acoustic ray trace program.